UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

CONTROL DE L'ENTORN EN ANIMALS

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Títol: Control de l'entorn en animals

Autora: Carlota Balcells Nuevo

Director: Javier Fernández García

Ponent: Enric Mayol

Data: 27-09-2016

ÍNDEX

1 Introducció al projecte

1.1 Motivació

Avui en dia, és cada cop més comú tenir animals estranys a casa: serps, eriçons, dracs barbuts, etc. El que caracteritza a aquests animals, és que no estan acostumats a segons quins medis climàtics, pel que els hem de tenir tancats en terraris i controlats a nivell de temperatura. Alguns fins i tot també d'humitat.

La majoria de gent que té algun d'aquests animals a casa no els hi queda més remei que fer ús del DIY (*Do It Yourself*), i comprar tots els aparells electrònics necessaris en alguna web, per muntar després algun mecanisme de control, com per exemple un termòstat.

Els termòstats actuals ens venen ja amb un sensor de temperatura. L'únic que hem de fer és les connexions amb la bombeta de calor i/o rajos ultraviolats.

A l'any 2005, a l'institut IVRAE, Massimo Banzi va inventar *Arduino*, una placa de desenvolupament, per facilitar als estudiants un medi per poder aprendre per sí mateixos, de forma més fàcil i sense que fos massa car.

En aquell moment les plaques eren molt cares, i aquests primers *Arduinos* es van vendre a 1€.

El que volem fer en aquest projecte és utilitzar un *Arduino* per un propòsit similar:

- Simular un termòstat: Capturar valors de temperatura i actuar en funció d'aquest sobre un *led*, que ens simularà un aire condicionat/calefacció.
- Capturar també els valors d'humitat, llum i l'estat d'unes portes magnètiques.
- Poder establir dos llindars de valors, en el qual si un sensor arriba a aquell valor, es crearia una alarma (prèvia o greu, com veurem més endavant).
- Poder reaccionar davant aquestes alarmes de diverses formes: Rebre un correu, un missatge, una trucada telefònica, o inclús fer que s'activi/desactivi un aparell. En cas que ja hi hagués una alarma del nivell generat per aquell sensor, no es faria res.
- Poder validar les alarmes mitjançant un comentari.
- Veure el registre històric d'alarmes.

- Veure en un gràfic les dades d'un sensor determinat entre dos rangs de dates.
- Veure en una pantalla tots els valors dels sensors d'una ubicació.
- Si en la pantalla anterior cliquem en el valor d'un sensor, que ens mostri en un gràfic les dades d'aquell sensor en les últimes 6 hores.
- Poder configurar tots els valors de la *bbdd* des d'un *backoffice*: microcontroladors, sensors, tipus de mesures...
- Tenir usuaris amb diferents tipus de menús, per lo tant rols diferents del *backoffice*.
- Tenir un registre de tots els canvis que s'han produït a nivell de *backoffice*:
    - S'ha insertat un nou valor, eliminat o modificat? (I – insert, U – Update, D – delete).
    - A quina taula s'ha produït.
    - Quin camp s'ha tocat.
    - El identificador o clau primària d'aquella taula.
    - El valor del identificador o clau primària d'aquella taula.
    - El camp que s'ha modificat en cas que sigui una modificació.
    - El seu valor antic en cas que sigui una modificació.
    - El seu valor nou en cas sigui una modificació.
    - Data del canvi.
    - Per quin usuari ha sigut.

1.2 La empresa: a què es dedica?

Treballarem per una empresa que fa recerca per indústria farmacèutica (concretament és una empresa preclínica que experimenta amb animals), i es dedica a criar i fer experiments amb gossos beagle, gats, fures i conills, alguns lliures de vacunes.

Aquest experiments es realitzen en sales blanques. Les sales blanques són sales que s'utilitzen molt en l'àmbit farmacèutic, ja que són sales en les quals la contaminació és molt baixa o quasi inexistent. Necessitem l'ús d'aquestes sales per a que els animals no es contaminin i estiguin en condicions òptimes (temperatura, humitat, etc). Com són sales controlades, no podem obrir una porta que posi en contacte

una sala blanca directament amb l'exterior, ja que aquesta es contaminaria automàticament. Per tot això necessitem un control de les portes magnètiques.

1.3 Objectius

L'objectiu d'aquest projecte és bàsicament poder controlar l'entorn dels recintes on tindrem aquests animals mitjançant l'ús de plaques *Arduinos* i sensors de varis tipus: temperatura, humitat i llum per les sales, i els estats de les portes magnètiques per a que aquestes no es contaminin.

Per a això, evidentment, cal estudiar la tecnologia que s'utilitzarà en els *Arduinos*. Com connectar-los els sensors necessaris, capturar els valors d'aquests, configurar-los la xarxa, connectar-los al *webservice* on enviaran les dades, etc. El llenguatge que utilitzarem per programar-los ja ens ve definit pels mateixos *Arduinos*: C.

A més a més, cal estudiar també quina tecnologia utilitzarem per desenvolupar el *backoffice* i en base a aquest, quin llenguatge de programació utilitzarem.
Per últim, en referència a les accions a realitzar, rebre les dades del microcontrolador, mostrar en gràfics els valors dels sensors, etc..., caldrà estudiar quines llibreries necessitarem per implementar-ho tot.

Després es realitzarà la especificació del projecte, disseny i desenvolupament de l'aplicació i les proves corresponents.

A mode de resum, a continuació, s'indiquen les tasques en les que hem dividit el projecte:

- Estudi de les diferents tecnologies utilitzades al projecte.
- Estudi del llenguatge de programació per a desenvolupar el *backoffice*.
- Estudi de les diferents llibreries necessàries per implementar els *webservices*.
- Disseny del *backoffice*.

- Desenvolupar el *backoffice*.
- Muntatge dels microcontroladors i sensors.
- Programar el codi d'aquests.
- Introducció de dades des del *backoffice*.
- Desenvolupar els *webservices* necessaris.
- Provar els diferents blocs que tindrem: microcontrolador, backoffice i webservices.
- Implementació dels triggers per a tenir un control de tots els canvis.

Al programar el microcontrolador s'ha de tenir en compte:

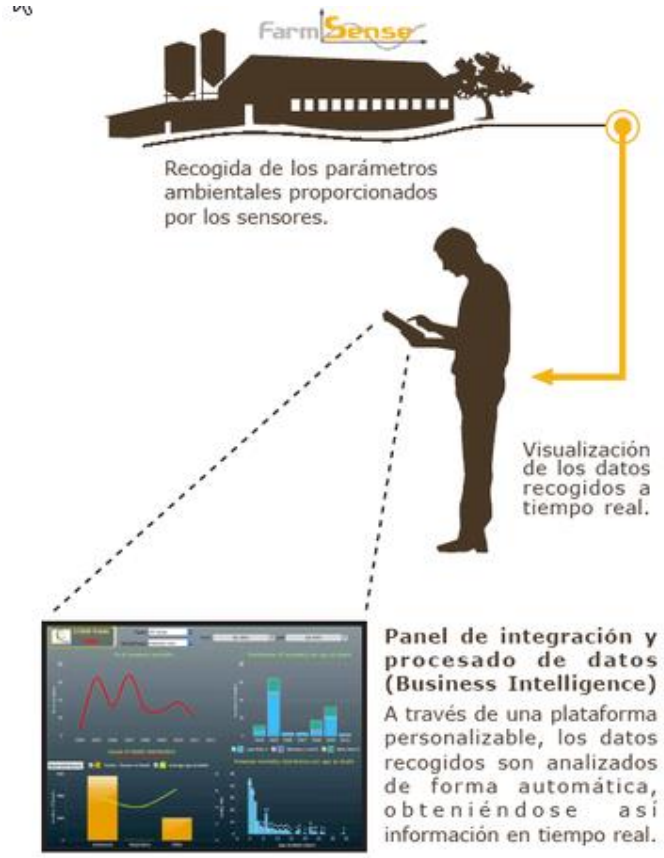
- Connexió a la xarxa per l'enviament de dades.
- Com rebre dades dels sensors.
- Com connectar-nos a un *webservice* i enviar aquestes dades.
- Com controlar un *led* mitjançant un relé.

2 Estat de l'art

2.1 Projectes similars

Alguns dels projectes que es fan amb sensors junt amb un microcontrolador, són relacionats amb alarmes de seguretat: detectar moviments, llums i so. Ens podem construir un sistema d'alarmes a la nostra casa, no molt segur però sí que ens avisi dels possibles atacs.

Un projecte ja existent amb sensors, és FarmSense, un sistema de recollida, monitorització i anàlisis de paràmetres ambientals orientat a activitats ramaderes. Amb aquest sistema el que fan és tenir un control sobre les seves instal·lacions, millorant les condicions de manteniment i sanitat, disminuint així els costos de produccció (1).



Con FarmSense podrá:

- Verificar el correcto funcionamiento de sus sistemas de control ambiental
- Establecer alarmas para detectar cualquier incidencia (roturas de bebederos, fallos en el sistema eléctrico, temperaturas extremas y muchos otros).
- Ajustar de forma precisa las dosis de medicación, evitando derroches innecesarios
- Minimizar el gasto energético de sus instalaciones
- Establecer alarmas sanitarias
- Recibir informes periódicos para optimizar el manejo y, por tanto, la producción.

Una altra cosa en comú amb el nostre projecte, és que a més de tenir un control visual ambiental, també poden establir alarmes per detectar alguna incidència.

Un altre projecte que també conté sensors i animals, seria el projecte SERIDA. Aquest projecte el que té com a finalitat és el control qualitatiu dels aliments que els donaran a uns animals. Es controla tant la qualitat del menjar, com la ració que se'ls dóna (2).

En quant a projectes de control de qualitat d'aliments on els sensors s'hi vegin involucrats, hi ha molts. Agrodigital seria un d'ells. Utilitzen sensors sense fils per tenir un control de qualitat i poder supervisar els productes hortofructícols durant el seu transport i distribució. El més important en el seu transport, per ells, és que es mantingui la temperatura de fred constant des que surten del mans del productor fins el consumidor final (3).

2.2 Microcontrolador

2.2.1 Definició?

Un microcontrolador és com un ordinador petit, en un sol circuit integrat o placa, format per una cpu, una memòria i unes entrades i sortides analògiques i/o digitals que ens permeten controlar dispositius i sensors (4). La font d'alimentació d'aquests controladors sol ser externa.

La memòria dels microcontroladors ens permeten instal·lar-los-hi un programa o sketch, o sigui, ens permeten programar mitjançant codi com la placa controlarà els seus dispositius.

2.2.2 Quin hem escollit i per què?

L'elecció del microcontrolador és important, ja que segons quin escollim tindrem un mètode u altre per carregar el programa, i les possibilitats de connectar-nos a mòduls externs i ser capaços d'entendre's entre sí.

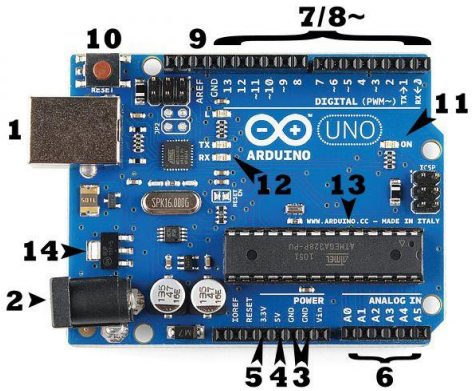
Actualment existeixen molts tipus d'aparells electrònics amb sensors integrats, que ens permeten captar lectures de valors com: temperatura, consums elèctrics, pressió atmosfèrica, potència, llum, etc. No obstant això, és difícil trobar un que sigui capaç de centralitzar totes aquestes mesures i sobretot, dur un control, o oferir accés remot a elles, a baix cost.

Avui en dia totes les empreses volen productes de qualitat, que els doni les prestacions que necessiten de forma eficient però sense deixar de banda el preu econòmic. Volen baix cost i mínim manteniment.

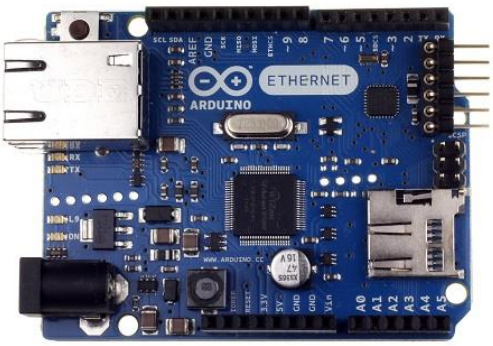
Per aquests motius hem escollit *Arduino*. És un microcontrolador potent, a baix cost, amb una disposició gran de pins varis: analògics i digitals, que a més a més disposa d'una gran varietat de mòduls que podem afegir a la placa principal i créixer en funcionalitats.

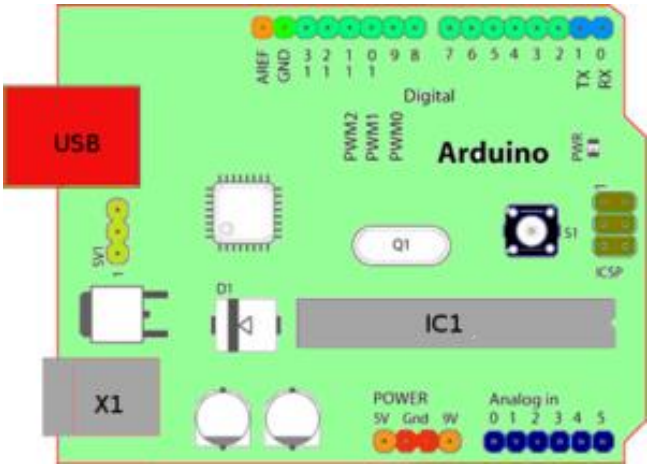
Sota podem veure com és una placa *Arduino* de tipus UNO (nosaltres utilitzarem un *Arduino Ethernet*, més bàsic però molt similar), que és la més utilitzada, i veiem també les diferents parts de les que està compost.

Arduino UNO



Arduino Ethernet utilitzat





Potencia - USB (#1 en imatge) / Conector de Adaptador (#2 en imatge)

Cada placa *Arduino* necessita una forma de ser alimentada elèctricament. Aquesta pot ser alimentada des de un cable USB que ve del ordinador o un cable de corrent elèctrica amb el seu adaptador. La connexió USB és també com carregarem el codi a la placa *Arduino*.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Els pins de la placa *Arduino* és on es connecten els cables d'un circuit. *Arduino* té varis tipus diferents d'entrades, cada una de les quals està marcada en el taulell i que s'utilitzen per diferents funcions:

• **GND (#3 en imatge):** Abreviatura de "terra" (en Angles). Hi ha varis pins GND en el placa *Arduino*, qualsevol dels quals poden ser utilitzats per a connectar a terra el circuit.

• **5V (#4 en imatge)** i **3.3V (#5 en imatge):** Són els subministres pin 5V 5 volts d'energia, i els subministres de pin 3.3V 3.3 volts de potència.

• **Analògic (#6 en imatge):** L'àrea de pins en el marc del 'analògica' etiqueta (A0 a A5) són analògics. Aquests pins poden llegir la senyal d'un sensor analògic (com un sensor de temperatura) i convertir-lo en un valor digital que podem llegir.

• **Digital (#7 en imatge):** Són els pins digitals (del 0 al 13). Aquests pins es poden utilitzar per la entrada digital (si es prem un botó) i sortida digital (com encendre un LED).

• **PWM (#8 en imatge):** El caràcter (~) al costat d'alguns dels pins digitals (3, 5, 6, 9, 10 i 11) vol dir que aquests pins actuen com a pins digitals normals, però a més a més també es poden utilitzar per una cosa anomenada modulació d'ample de pols (també conegut com PWD, sigles que provenen del anglès *pulse-width modulation)*.

• **AREF (#9 en imatge):** Suports de referència analògica. La majoria de les vegades es pot deixar aquest pin sol. A vegades s'utilitza per establir una tensió de referència externa (entre 0 i 5 volts) com el límit superior per als pins d'entrada analògica.

Botó de reinici (#10 en imatge)

Prement aquest botó es connectarà temporalment el pin de *reset* (reinici) a terra i reiniciarà qualsevol codi que es carregui al *Arduino*.

Indicador LED d'alimentació (#11 en imatge)

Aquest LED s'ha d'encendre cada vegada que es connecti la placa *Arduino* a una presa elèctrica. Si aquesta llum no s'encén, hi ha una alta probabilitat de que alguna cosa estigui malament.

LEDs RX TX (#12 en imatge)

TX és la abreviació de transmissió, RX es la abreviació de recepció. Aquestes marques poden apareixen en la electrònica per indicar-nos els responsables de la comunicació en sèrie. En el nostre cas hi ha dos llocs en la placa *Arduino UNO* on apareixen TX i RX - una vegada en els pins digitals 0 i 1, i una segona vegada junt als indicadors LED de TX i RX (12). Aquests LEDs ens donaran indicacions visuals quan la nostra placa *Arduino* estigui rebent o transmetent dades (com quan estem carregant nou codi en la placa).

Microcontrolador (#13 en imatge)

La part negra amb totes les potes de metall, és un circuit integrat (IC, per les seves sigles en Anglès). És com el cervell del nostre *Arduino*. El tipus de placa és important, ja que necessitarem saber el tipus de IC, junt amb el tipus de targeta, abans de carregar un nou programa des del software de *Arduino*.

Regulador de Voltatge (#14 en imatge)

El regulador de voltatge fa exactament lo que diu - controla la quantitat de tensió que es deixa en la placa *Arduino*. Pensa en ell com una espècie de guardià; donarà l'esquena a una tensió addicional que podria danyar el circuit. Per suposat, té els seus límits, per lo que no s'ha de connectar *l'Arduino* a res superior a 20 volts (5).

A *Arduino* hi podrem connectar qualsevol tipus de sensor, sempre hi quan tinguem pins (connexions físiques) disponibles.

A més a més, amb el mòdul *d'Ethernet*, podrem connectar-nos al *webservice* ubicat en un servidor extern, per poder enviar-li les mesures dels seus sensors.

També, ens permet actuar sobre un pin, de forma externa, si ho necessitéssim.

L'hem comparat amb altres microcontroladors com PIC, que és un microcontrolador de Microchip molt popular fins fa uns pocs anys, que té moltes més aplicacions degut a que du més temps en el mercat que *Arduino*.

En resum, hem escollit una placa *Arduino* per:

1) Té moltíssims mòduls per poder fer pràcticament tot el que se'ns pugui ocórrer, tenint en compte les limitacions de memòria i les connexions de la placa.
2) Es programa en C o pseudo-C, amb llibreries externes especials per *Arduino*
3) És *OpenSource*, amb el que tots els mòduls o components que anem necessitant seran de baix cost.
4) En comparació amb altres microcontroladors com PIC, el microcontrolador en el que està basada la placa de desenvolupament *Arduino* (*Atmega328*) és més ràpid i fàcil d'aprendre a utilitzar, més barat, i de codi obert (6).

2.3 Sensors

2.3.1 Què són?

Primer de tot començarem definint què és un **transductor**. Un transductor és qualsevol dispositiu capaç de convertir un tipus de magnitud física, habitualment energia, en un altre tipus de magnitud física. La majoria converteixen energia elèctrica en un desplaçament mecànic, o bé converteixen una magnitud física no elèctrica, (com ara temperatura, llum o so) en un senyal elèctric.

Un **sensor** és un dispositiu capaç de mesurar magnituds físiques o químiques, anomenades variables d'instrumentació (per exemple: temperatura, intensitat lumínica, distància, acceleració, inclinació, desplaçament, pressió, força, torsió, humitat, pH, etc), i transformar-les en variables elèctriques (una resistència elèctrica, una capacitat elèctrica com en un sensor d'humitat, una tensió elèctrica com en un termoparell, un corrent elèctric com en un fototransistor, etc). Un sensor sempre està en contacte amb la variable de instrumentació i l'adapta per a que el pugui llegir un altre dispositiu. En el nostre cas, *l'Arduino*.

Un exemple per a que entenguem millor el seu significat, seria comparant-lo amb un termòmetre de mercuri. El termòmetre de mercuri aprofita la propietat que té el mercuri de dilatar-se o contraure's segons la temperatura. Un sensor també es pot dir que és un dispositiu que converteix una forma d'energia en una altra.

Un cop hem vist què és un transductor i què és un sensor, veiem que un sensor és un tipus de transductor que transforma la magnitud que es vol mesurar o controlar, en una altra que facilita la seva mesura. Poden ser de indicació directa (un termòmetre de mercuri) o poden estar connectats a un indicador (possiblement a través d'un convertidor analògic o digital, un computador i un visualitzador) de forma que els valors detectats poden ser llegits per un humà.

A diferència del transductor, el sensor sempre està en contacte directe amb la variable que es vol mesurar o controlar.

Hi ha dues maneres de classificar els sensors:

- Sensors analògics: Un sensor analògic és aquell que com a sortida emet un senyal comprès per un camp de valors instantanis que varien en el temps, i són proporcionals als efectes que s'estan mesurant; per exemple, un termòmetre és un dispositiu analògic. La temperatura es mesura en graus que poden tenir, en qualsevol moment determinat, diferents valors que són proporcionals al seu indicador, o a la seva "sortida" en cas d'un dispositiu electrònic.(7)

  A grans trets té més qualitat que els digitals, però també resulta molt més car. (8)

- Sensors digitals: Són aquells que davant un estímul poden canviar d'estat ja sigui de 0 a 1, o de 1 a 0 (parlant en termes de lògica digital) en aquest cas no existeixen estats intermedis i els valors de tensió que s'obtenen són únicament dos, 5V o 0V (o valors molt pròxims). Per exemple, una caixa que és transportada arriba al final d'un recorregut i activa un sensor digital; llavors, la senyal 0 del sensor en repòs, canvia immediatament a 1, donant conte al sistema de tal condició..... (7)

Pel fet de donar la informació en estats (0,1) és clar que perdem informació, però surt més econòmic (8).

Actualment existeixen moltes possibilitats per poder mesurar paràmetres d'ambient i poder utilitzar-los en benefici propi, per a un estudi, per poder controlar l'entorn o simplement per tenir-ne un coneixement.

A nosaltres ens han demanat mesurar lo següent:

- Temperatura
- Humitat
- Lluminositat en sales
- Estat de les portes dels laboratoris

Però podríem tenir molt més com :

- Pressió Atmosfèrica
- Potència
- Concentració de CO2
- NH3 Amoníac

2.3.2 De temperatura

Els sensors de temperatura són dispositius que transformen els canvis de temperatura en canvis de senyals elèctriques, que són processats per equip elèctric o electrònic.

Hi ha tres tipus de sensors de temperatura: termistors, RTD i els termoparells.

Un sensor de temperatura comunament està format per l'element sensor, de qualsevol dels tres tipus anteriorment esmentats, una baina que l'envolta i que està plena de un material molt conductor de la temperatura, per a que els canvis es transmetin ràpidament al sensor, i pel cable al que es connectarà el equip electrònic.

RTD (Resistance Temperature Detector):

És un sensor de temperatura basat en la variació de la resistència d'un conductor amb la temperatura. Si la temperatura augmenta o disminueix, la resistència també augmenta o disminueix en la mateixa proporció (9).

2.3.3 D'humitat

Hi ha diferents tipus de sensors d'humitat, els més usuals són:

- Mecànics: aprofiten els canvis de dimensions que pateixen certs tipus de materials en presència d'humitat, com per exemple: fibres orgàniques o sintètiques...

- Basats en sals higroscòpiques: dedueixen el valor de la humitat en l'ambient a partir d'una molècula cristal·lina que té molta afinitat amb l'absorció de l'aigua.

- Per conductivitat: la presència d'aigua en un ambient permet que a través d'unes reixetes d'or circuli un corrent, ja que l'aigua és bona conductora de corrent. Segons la mesura de corrent es dedueix el valor de la humitat.

- Capacitius: es basen senzillament en el canvi de la capacitat que pateix un condensador en presència d'humitat.

- Infrarojos: aquests disposen de dues fonts infraroges les quals absorbeixen part de la radiació que conté el vapor d'aigua.

- Resistius: estan basats en la detecció del canvi de resistència produïda en augmentar la humitat. La resistència ajustable serveix per controlar manualment el límit d'humitat o llinda de funcionament del sensor. Podem ajustar la sensibilitat del dispositiu, actuant sobre aquesta resistència.

En el nostre projecte, utilitzarem un sensor digital de tipus SHT15, que és un sensor 2 en 1. Ens dona valors de temperatura i humitat en un sol xip. Així ens estalviem usar més pins de la placa *d'Arduino*, que podrem aprofitar per altres finalitats (10).

2.3.4 De llum: LDR

La LDR és un dispositiu electrònic semiconductor emprat per produir un senyal de sortida en resposta a un altre senyal d'entrada (una resistència) que té la propietat de variar el seu valor en funció de la intensitat de la llum incident. En presència d'una gran intensitat de llum la seva resistència elèctrica és baixa, de l'ordre dels 100 Ω. En canvi a les fosques, sense llum, la seva resistència és alta, de l'ordre de 10 MΩ (11).

2.3.5 Porta magnètica

Es el tipus més senzill de detector, però a la vegada més efectiu. Es tracta d'un dispositiu format per dos parts:

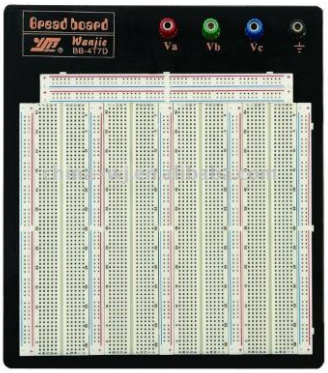
1. Encapsulat amb *reed switch*: càpsula de vidre que conté dos contactes metàl·lics i un parell de terminals que permeten accedir a connectar aquests dos contactes. Els dos contactes normalment estan aïllats l'un de l'altre. Quan un camp magnètic de la magnitud adequada s'apropa, aquests contactes es tanquen.

2. Encapsulat amb imant: És simplement un material magnetitzat (12).

2.4 Protoboard

La placa de proves, també coneguda com a P*rotoboard* o *breadboard*, és una placa d'ús genèric reutilitzable o semi permanent, usada per construir prototips de circuits electrònics amb soldadura o sense. Normalment s'utilitzen per a la realització de proves experimentals.

En aquest projecte farem ús d'una *Protoboard*, ja que estem en mode experimental, de proves. Així ens estalviem soldadures (13).



2.5 Relé

Un relé (del francès *relais*, relleu) és un mecanisme elèctric, inventat per Joseph Henry el 1835, que permet modificar l'estat d'un commutador elèctric gràcies a l'electricitat. A grans trets és un commutador elèctric que és accionat per un electroimant que obre o tanca un o diversos contactes. Una característica important d'aquest component és que permet controlar circuits elèctrics de voltatge o intensitat molt superior al d'entrada (14).

Nosaltres en farem ús per controlar un led, que simularia un sistema d'aire condicionat. El connectarem a *l'Arduino*, i al relé connectarem el led.

2.6 Scriptcase

Scriptcase Generador de PHP és una eina poderosa per augmentar la productivitat del desenvolupament web, estalviant temps i augmentant els beneficis.

Pot construir sistemes complets i crear informes personalitzats segurs i ràpids. És la millor i més eficient eina de desenvolupament web del mercat.

Com bé indica el seu nom, està basada en PHP, i ens permetrà desenvolupar tot el sistema de *backoffice* intern, reduint el temps de desenvolupament fins i tot en el 80%.

ScriptCase pot treballar en qualsevol navegador web, ja sigui en la nostra xarxa local o mitjançant internet, i permet que varis desenvolupadors puguin treballar simultàniament en el mateix projecte. Tants com llicències tinguem contractades.

Sols hem de connectar la base de dades que tinguem feta pel projecte (MySQL, Oracle, SQL Server, ...) per generar aplicacions que s'executin independentment de *ScriptCase*, i pot ser publicat en qualsevol servidor web que tingui PHP (versió 5.5 màxim) instal·lat (15).

A l'apartat del programari utilitzat podem veure on aconseguir-lo, i funciona amb els sistemes operatius: Windows, Linux/Unix y Mac OS X.

2.7 Asterisk

Asterisk és una aplicació de programari lliure (sota llicència GPL) que proporciona funcionalitats d'una central telefònica (PBX, ***P**rivate **B**ranch E**x**change*). Com qualsevol PBX, es pot connectar un nombre determinat de telèfons per a fer cridades entre si i fins i tot connectar a un proveïdor de VoIP o bé a una RDSI tant bàsics com primaris.

Actualment és el líder mundial en plataformes de telefonia de codi obert. Es un software que pot convertir un ordinador de propòsit general, en un sofisticat servidor de comunicacions VoIP.

Aquest sistema de centraleta IP, es utilitzat per empreses de totes les mides, per millorar la seva comunicació, incloent empreses com Google, Yahoo, IBM, e inclús l'exercit dels EE.UU.

Actualment les solucions *Open Source* representen el 18% de les centraletes telefòniques instal·lades en tot el món (segons el Eastern Management Group) i Asterisk és el líder en el mercat de codi obert de centraletes VoIP (VoIP PBX) (16).

Nosaltres ja tenim en un servidor un sistema *Asterisk* muntat, que aprofitarem per connectar-nos a ell i poder fer trucades a dispositius mòbils.

2.8 Plataforma SMS

Utilitzarem una plataforma ja existent d'enviament de missatges. Ens connectarem amb ells a través de *curl*, amb un identificador de client, passant-li tota la informació del missatge a enviar.

La plataforma es diu Sit Consulting.

2.9 Altres

2.9.1 Phpmailer

Com tenim un tipus d'acció per notificar-nos que tenim una alarma activa en un sensor que és enviar un correu, necessitarem d'alguna eina per enviar correus.

En aquest projecte ens hem decantat per utilitzar la llibreria més clàssica i utilitzada en PHP per enviar correus: *Phpmailer*.

L'hem escollit perquè :

- Volíem una eina de codi obert (aquesta té llicència LGPL 2.1), que no fos la clàssica funció de *php* mail(). Aquesta última no proveeix cap tipus d'ajuda per fer ús d'eines com correus en HTML i poder adjuntar documents en un correu.
- Probablement és la llibreria més utilitzada per enviar correus en PHP.
- És utilitzada en molts projectes Open Source com: WordPress, Drupal, Jooma!...
- Integra suport STMP
- Podem enviar correus amb múltiples destinataris a TOs, CCs, REPLY-TOs..
- Té alternatives per clients que no puguin llegir correus en Html.
- Suporta contingut amb codificació utf8, 8bit, base64, binary...
- Autenticació smtp amb LOGIN, PLAIN, NTLM, CRAM-MD5 i Google's XOAUTH5 mecanismes sobre SSL i transport TLS.
- Missatges d'error en 47 idiomes.
- Suport per autenticacions DKIM i S/MIME.
- Compatible amb PHP 5.0 i posteriors.
- I molt més (17).

2.9.2 RGraph

Per les gràfiques de dades dels sensors, tant les de 6 hores com les de entre dos períodes de temps, necessitarem d'una eina que ens ajudi a veure en un gràfic aquesta informació.

RGraph és una llibreria de gràfics JavaSript interactiva que construeix els gràfics dinàmicament amb JavaScript i amb la etiqueta canvas de HTML5. Es poden generar molt tipus de gràfics diferents per tot tipus de visualitzacions. Des de gràfics de barres i línies fins a barres progressives.

A més té les següents característiques:

- És fàcil d'utilitzar.
- És llicència GPL permissiva.
- Podem generar fins a 50-60 tipus de gràfics diferents.
- Suport en gràfic s 2D i 3D.
- Gran suport de dispositius.
- Excel·lent funcionament.
- Eines per importar fitxers CSV i fulles de càlcul de Google.
- Extensa documentació, exemples i recursos per aprendre.
- Gran suport per part del equip RGraph (18).

rgraph
Interactive JavaScript charts for your website

3 Planificació

3.1 Fases

La gestió del projecte s'ha dividit en les fases típiques d'inici, planificació, execució i tancament. Durant l'execució de totes les fases s'anirà completant la memòria del projecte.

3.1.1 Fase inicial

3.1.1.1 Definició del producte

Primeres converses amb l'empresa per determinar les funcionalitats del *backoffice* i de *l'Arduino*. Fer un anàlisis de requisits inicial i decidir quines tecnologies utilitzarem.

3.1.1.2 Formació

Trobar els manuals i documentació necessària per a prova les diferents tecnologies utilitzades:

- Arduino
- Scriptcase
- Asterisk
- Llibreries a utilitzar

3.1.2 Fase de planificació

En aquesta fase és on es fa la planificació, estimació d'hores i especificació del projecte.

3.1.3 Fase d'execució

Disseny i implementació del projecte. En aquesta fase s'aniran realitzant certes iteracions per a veure amb el client que, tant el disseny com implementació de les

diferents funcionalitats del projecte, compleixen els requeriments. Seguirem un model de desenvolupament en cascada. Aquest model és un procés en seqüència, fàcil de desenvolupar, en el que el passos són vistos cap a baix (com una cascada d'aigua) a través de les fases d'anàlisis de les necessitats, el disseny, implementació, proves (fins la validació), integració i manteniment (19).

A cada iteració s'inclouran les funcionalitats a implementar, de la següent manera:

Iteració	Funcionalitats
1	Muntar tots els sensors. Programar el microcontrolador configurant-lo per a que es connecti a internet i que reculli les dades dels sensors que hem muntat cada X segons per enviar-les.
2	*Backoffice*: iniciar sessió, creació microcontroladors, sensors, mesures, alarmes, accions, veure historial alarmes, validar-les, creació usuaris.
3	Implementació *webservice* per rebre les dades dels sensors. Veure que s'estan enviant correctament Que podem visualitzar les dades des del *backoffice* Que es llencen les alarmes si els valors estan entre els llindars establerts
4	Implementació de les accions varies. Enviar correus. Enviar missatges amb la plataforma sit Mobile. Trucar a telèfons connectant-nos amb el servidor *Asterisk*.
5	Muntar led connectat a relé. Modificar microcontrolador per a que activi/desactivi el led.
6	Implementar pantalla per veure les dades dels sensors d'una ubicació. Implementar pantalla per veure en un gràfic les dades de les últimes 6 hores d'un sensor determinat. Fer una pantalla similar per rangs de dates de qualsevol sensor.
7	Implementar els triggers en totes les taules que volem tenir-hi un control.

3.1.4 Fase de tancament

3.1.4.1 Últims retocs amb el client

Confirmar amb el client que el producte final obtingut correspon a la especificació acordada.

3.1.4.2 Revisió de la memòria

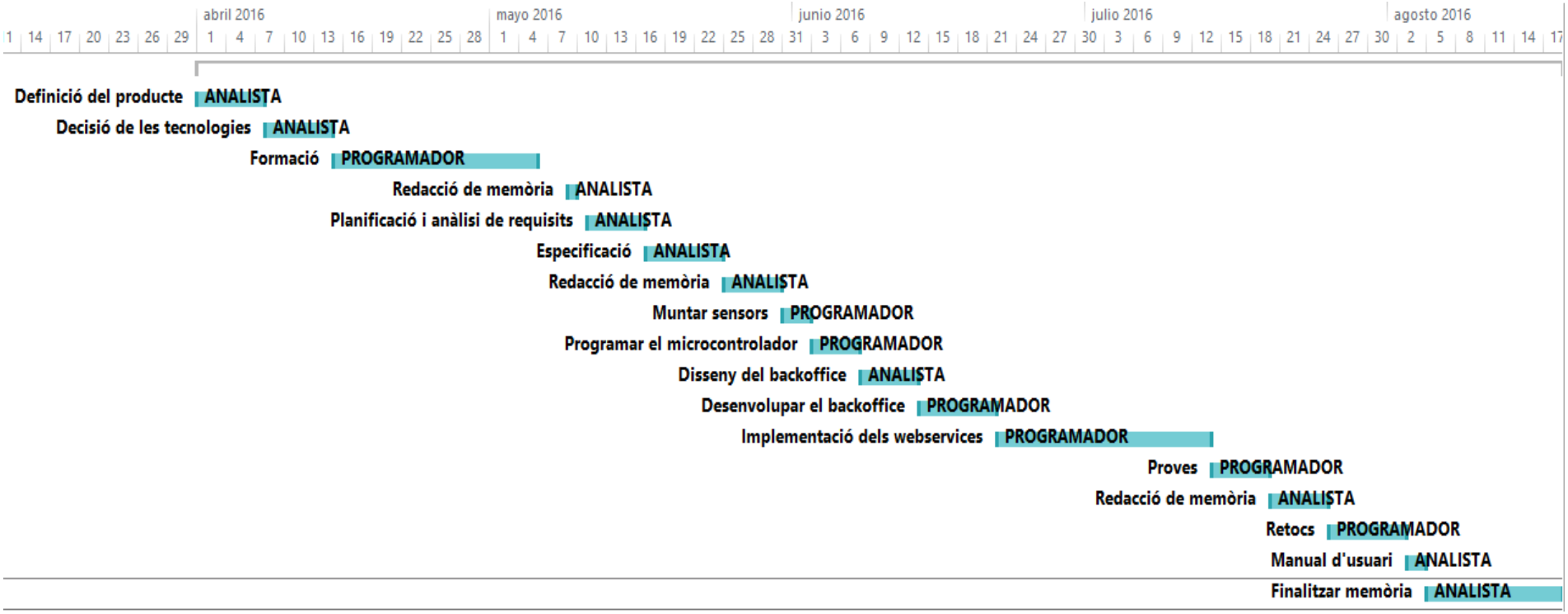
Acabar de redactar la memòria i documentació del projecte (com pot ser el manual d'usuari).

Estimació d'hores

Fase inicial (155h)		
	Definició del producte	25
	Decisió de tecnologies i compra material necessari	40
	Formació	75
	Redacció memòria	15
Fase planificació (85h)		
	Planificació i anàlisis de requisits	25
	Especificació	40
	Redacció memòria	20
Fase execució (295h)		
	Muntatge dels sensors	20
	Programació microcontrolador	30
	Disseny del backoffice i webservices	40
	Implementació backoffice	40
	Implementació webservices i triggers	100
	Proves	40
	Redacció memòria	25
Fase tancament (80h)		
	Retocs	30
	Manual d'usuari	10
	Finalitzar memòria	30
		Total: 605h

S'han utilitzat dos rols per a realitzar-lo: Analista i Programador. L' Analista s'encarrega de definir el producte, fer l'estudi, especificació i disseny, mentre que el Programador tot lo relacionat amb la implementació. En el Gantt s'ha tingut en compte que una mateixa persona ha fet els dos rols. Si haguessin sigut dues persones diferents s'hagués pogut optimitzar el temps, ja que mentre el programador es forma, l'analista estaria fent la planificació i l'especificació.

Diagrama de Gantt de les tasques, ja assignades als dos rols:



3.2 Anàlisi de riscos

Tot projecte té riscos, per això cal fer una previsió per a poder gestionar de manera segura i fer els canvis de manera controlada i a temps.

3.2.1 Previsió temporal inadequada

Al no haver treballat mai amb *Arduinos*, es possible que tant durant la fase de formació com a la d'execució tardem més de l'esperat.

Probabilitat: molt alta.

Impacte: menor si en la fase de formació ho tenim tot clar; crític altrament.

Solució: Assignar-hi més recursos si n'hi ha i limitar l'abast del projecte. Tornar a estimar les hores.

3.2.2 Canvis inesperats en la definició del producte

És possible que l'aplicació no estigui completament definida abans de començar amb la implementació i que es demanin fer alguns canvis.

Probabilitat: baixa.

Impacte: mitjà.

Solució: comprovar que es pot canviar l'abast sense una desviació important de la planificació, proposar una reunió per definir l'abast de manera completa i acceptar els canvis necessaris.

3.2.3 Interrupció del desenvolupament per causes externes

Per malaltia, operacions en l'oficina o servidors, problemes tècnics...

Probabilitat: baixa.

Impacte: baix.

Solució: gestionar la planificació temporal. Si la interrupció és de més d'una setmana i no es poden recuperar les hores, comprovar la limitació de l'abast.

3.2.4 Qualitat insuficient

El producte final no compleix els requeriments. Aquest risc es pot detectar al final de cada iteració.

Probabilitat: molt baixa

Impacte: crític

Solució: tornar a l'especificació i al disseny de la iteració afectada. Comprovar que no hi haurà desviacions importants en la planificació.

3.3 Anàlisis de costos

3.3.1 Recursos humans

Al ser un projecte final de carrera serà una sola persona la qui realitzarà els dos rols que tindrem, sota supervisió del tutor de la universitat. No obstant, s'ofereix una simulació dels costos amb un equip real, que es compondria per un analista i un programador.

Recurs	Salari
Analista	40€/h
Programador	20€/h

D'acord amb l'assignació de recursos humans en cada tasca i la planificació temporal, es pot fer un càlcul del cost de desenvolupar el projecte. El calendari té dies de 8h.

Programador	370hores	7400,00€
Redacció documents	25h	300,00€
Manual usuari	10h	200,00€
Implementació	190h	300,00€
Proves	40h	600,00€
Retocs	30h	400,00€
Formació	75h	1500,00€
Analista	245hores	9800,00€

Especificació	40h	800,00€
Redacció documents	20h	600,00€
Definició del projecte	25h	800,00€
Disseny de l'aplicació	40h	1600,00€
Decisió tecnologia	40h	600,00€
Finalitzar documents	30h	800,00€
Redacció documents	25h	400,00€
Planificació i anàlisis	25h	600,00€

Cost recursos humans = cost programador + cost analista = 7400€ + 9800€ = 17200€

3.3.2 Recursos materials

A més a més, hem de tenir en compte el cost del material utilitzat per muntar els sensors, junt amb el portàtil que utilitzarem per desenvolupar el projecte.

Recurs	Preu
Arduino	25€/h
Cablejat vari	7€/h
Sensor SHT15	17€
Sensor llum	3,5€
Sensor porta magnètica	7€
Relé	14,5€
Protoboard	20€
Led	0,5€
1 Llicència Scriptcase Enterprise Edition	563€
Paquet 500SMS Sit Consulting	36,3€
Portàtil Asus X554L	700€

Cost recursos materials = cost material = 1393,8€ (22)

3.3.3 Cost total

La suma de les hores de feina i els recursos materials: Cost = 17200€ (Cost recursos humans) + 1393,8€ (recursos materials) = **18593,8€.**

4 Anàlisi de Requisits

4.1 Abast

L'aplicació ha de permetre veure l'historial de valors que ens envien els sensors que tenim, definir els llindars d'alarmes d'aquests, veure l'historial d'alarmes produïdes, validar-les, veure de forma gràfica els seus valors, i rebre notificacions o actuar sobre algun tipus de mecanisme si aquests valors no són correctes.

Per poder utilitzar el *backoffice*, cal que l'usuari disposi d'una connexió a internet i d'un navegador.

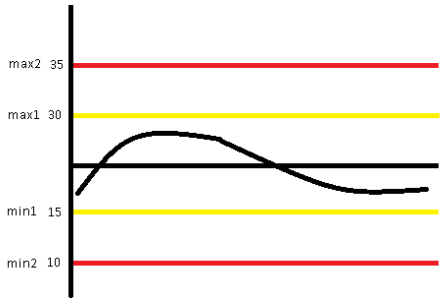
També cal tenir el microcontrolador sempre connectat a la xarxa, ja que sinó no podrà enviar els valors capturats pels seus sensors.

El primer cop que el microcontrolador envia informació al servidor, si no està donat d'alta, es crea automàticament.

El microcontrolador envia cada 50 segons el valor dels seus sensors a un servidor, que revisarà si aquests són correctes i actuarà en conseqüència.

Per saber si és correcta o no, abans haurem introduït des del *backoffice* uns valors mínims i màxims (explicat amb més detall a continuació) per cada sensor del qual ens interessi dur un control.

Per exemple, si una temperatura interior d'una gàbia on hi ha animals puja de 30 graus (ho definirem com a màxim1), volem rebre una notificació per tenir constància de que en aquella gàbia la temperatura està pujant i hem d'estar atents. Si arriba a pujar més de 35 graus (màxim2), volem rebre una altra notificació, ja que hauríem d'actuar. Passa el mateix amb temperatures que baixin de 15 graus (mínim1, primer avís), i de 10 graus (mínim2, avís greu).



Segons el que hem explicat, ara podrem entendre el que és una alarma i el que és una acció.

Una **alarma** seria definir uns límits màxims i mínims dels valors que ens doni un sensor. Si el valor del sensor està entre mínim1 i mínim2, o bé màxim1 i màxim2, l'alarma que es crearà serà de tipus lleu o avís previ. Si sobrepassa els llindars màxim2 o mínim2, l'alarma que es crearà serà de tipus greu.

Una alarma va lligada de forma única i exclusiva a un sensor. Només podem definir uns valors mínims i màxims per cada un d'ells.

Una possible ampliació d'aquest projecte seria jugar amb combinacions de valors de sensors diferents per generar alarmes.

Un cop tinguem les alarmes definides, els hi podem associar unes accions.

Una **acció** és el què hem de fer un cop salti una alarma. Tindrem varis tipus d'accions: enviar un correu, enviar un missatge al mòbil, trucar a un número de telèfon o bé activar/desactivar dispositiu.

Una alarma pot tenir múltiples accions, per exemple, per una alarma en el sensor de temperatura podem voler que s'activi l'aire condicionat, i que ens enviï un correu.

4.2 Requisits funcionals

L'aplicació de *backoffice* permet a l'usuari:

- Crear nous usuaris que puguin accedir a ella, amb un rol o un altre depenent del tipus d'accés que li vulguem donar.
- Veure en una pantalla intuïtiva el valor dels sensors d'una ubicació, i que al seleccionar un d'ells ens mostri en un gràfic el seu històric de les últimes sis hores.
- Veure el registre de tots els valors rebuts dels sensors.
- Veure en un gràfic els valors per un sensor entre varis rangs de dates.
- Gestionar les dades de:

  - Sensors
  - Tipus de sensors
  - Microcontroladors
  - Tipus de mesures
- Crear alarmes per a un sensor.
- Gestionar les alarmes: validar una alarma afegint-li un comentari, gestionar comentaris.
- Assignar a una alarma ja creada, una acció.
- Canviar l'idioma del *backoffice* en qualsevol moment.

4.3 Requisits no funcionals

- El microcontrolador ha de ser extensible per afegir-li nous sensors en un futur.
- Tant el codi del microcontrolador, com el de l'aplicació del *backoffice*, seran en anglès, amb els comentaris en català.
- El microcontrolador es connectarà a internet cada 50 segons, que serà el temps definit per a que enviï les dades al servidor. La resta del temps estarà sense connexió, capturant els valors dels sensors.
- L'aplicació ha de satisfer els factors de qualitat de canviabilitat i reusabilitat.

5 Especificació de l'aplicació

5.1 Model conceptual

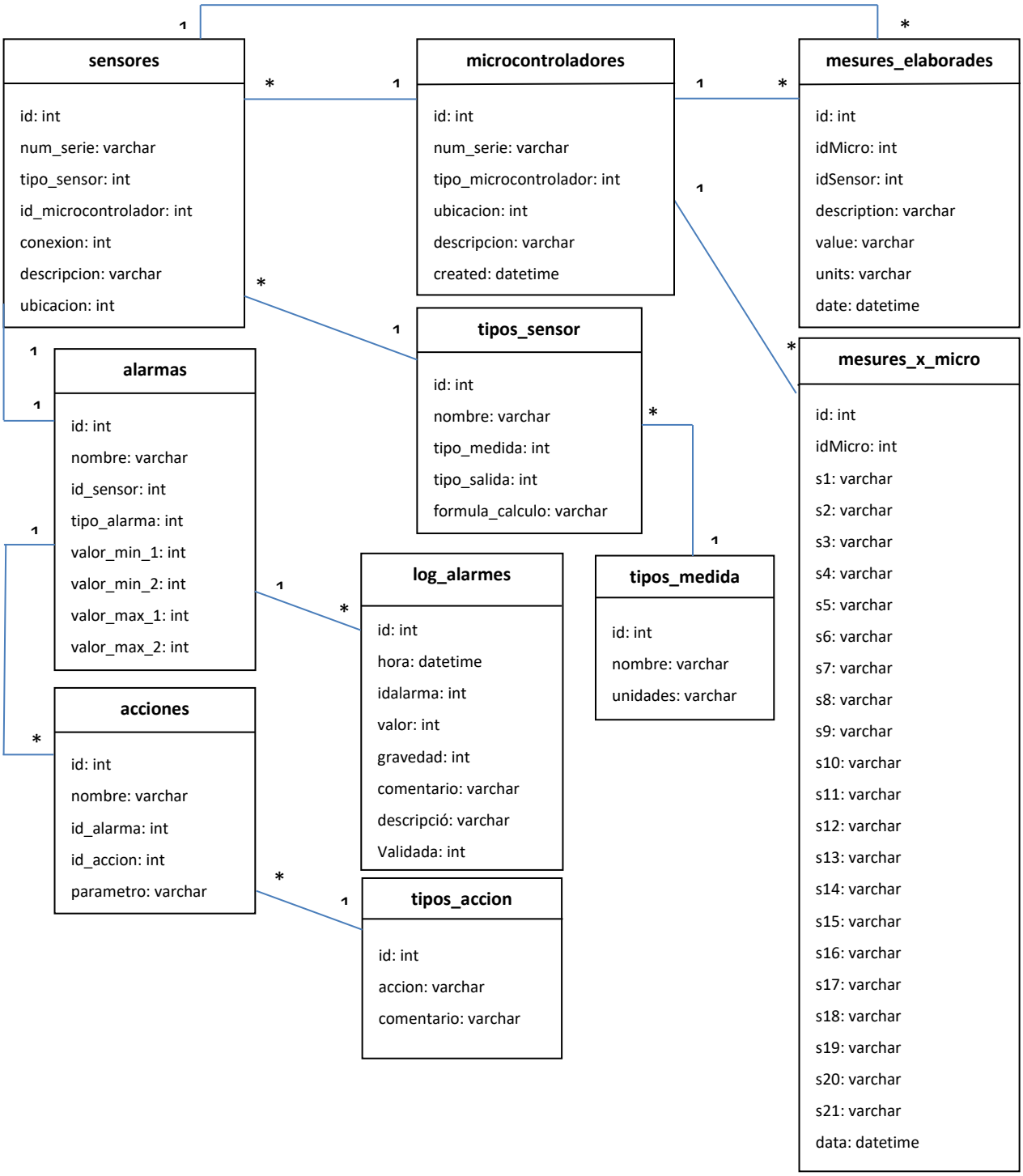
A continuació es mostra una representació dels conceptes del món real, no dels conceptes del programa, amb l'objectiu de comprendre millor el problema.

En el model conceptual es pretén explicar de manera visual els casos d'ús del sistema, tot allò que haurà de gestionar.

Si ens fixem en el diagrama que hi ha a continuació podem veure les restriccions d'integritat següents:

- Podem crear sensors i microcontroladors, però un sensor sempre tindrà un microcontrolador associat, en canvi un microcontrolador pot tenir varis sensors.
- Un sensor estarà format per un tipus de sensor, un microcontrolador al que pertany, una connexió (s1→s21) i una ubicació.
- Un tipus de sensor està relacionat amb la mesura que tindrà el sensor (watts, lúmens, graus, %...).
- Les dades de la taula "mesures_x_microcontrolador" seran les que rebrem directament sense tractar del codi programat de *l'Arduino*, i van associades a un microcontrolador. Un microcontrolador pot tenir moltes mesures, tantes com enviï el microcontrolador, que serà cada 50 segons.
- Les dades de la taula "mesures_elaborades" seran les dades que hem rebut de la taula "mesures_x_microcontrolador", un cop tractat cada connexió que té el microcontrolador i vist que té un sensor associat a aquella connexió. Cada mesura elaborada anirà associada a un microcontrolador i un sensor. Un sensor i microcontrolador poden tenir múltiples mesures.
- Un sensor pot tenir una única alarma, amb dos rangs de valors d'alerta lleu (min1 i max1) i dos rangs de valor d'alerta greu (min2 i max2).
- Per a una alarma podem tenir múltiples accions.
- Una acció tindrà un tipus d'acció (enviar correu, missatge...).
- Una mesura es registra en una data i hora determinada.
- Una alarma es llença a una data i hora determinada.

- No es pot generar una alarma d'un sensor si aquest ja té una alarma pendent de validar del mateix tipus de gravetat: lleu o greu.



5.2 Casos d'ús

En aquest capítol s'especifica la interacció entre actors i el sistema. Existeixen dos tipus d'actors:

- Usuari amb connexió
- Administrador amb connexió

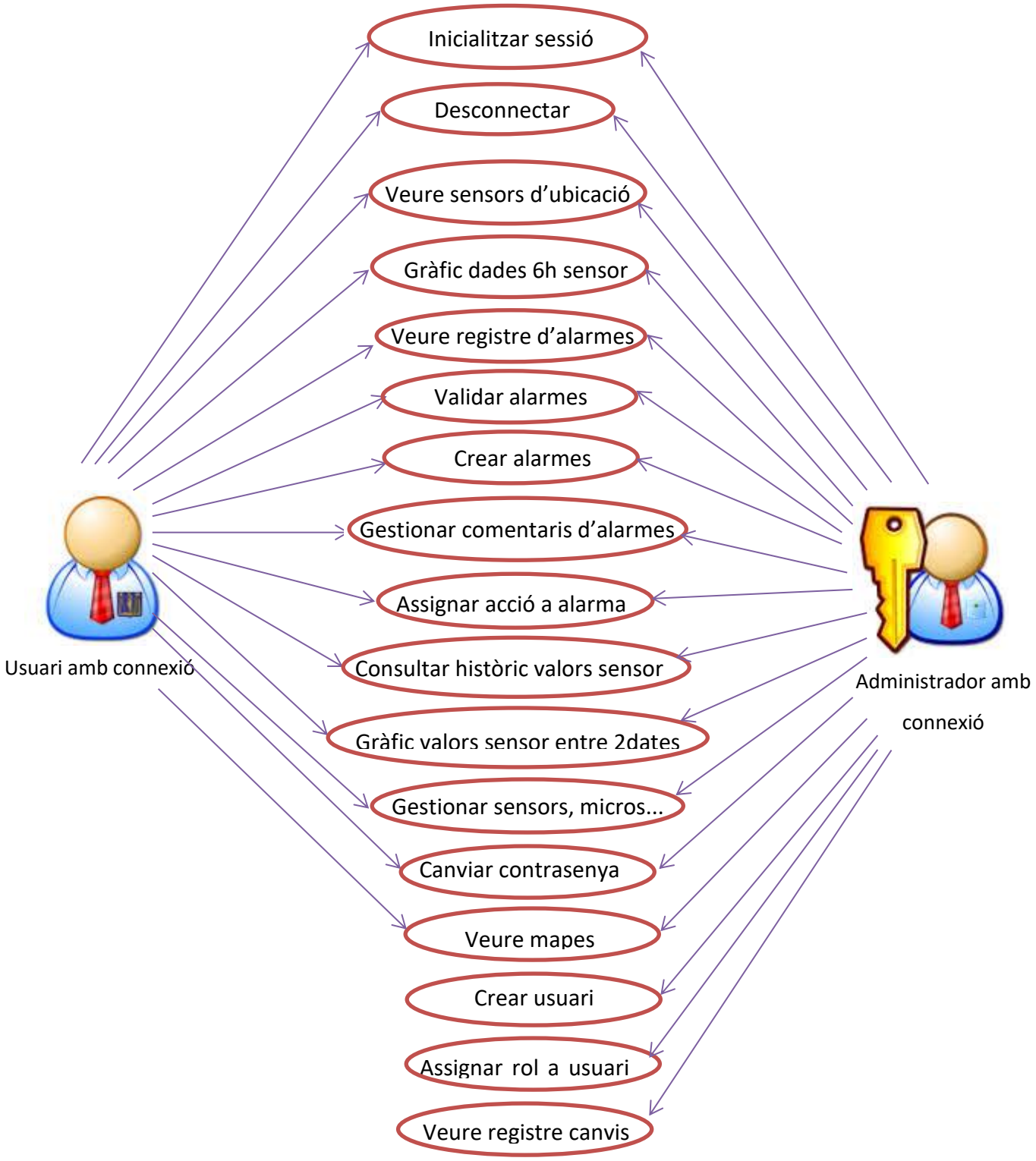
A continuació, es mostra el diagrama de casos d'ús, on s'hi pot veure quines són aquelles funcionalitats que tindrà l'usuari des del *backoffice*, sempre i quan tingui connexió:

- Inicialitzar sessió
- Veure dades sensors per ubicació. Cada mesura anirà assignada amb un color: verd si la dada es correcta, groga si la dada està en el primer nivell d'avís (entre min1 i min2 o entre max1 i max2), o vermella si està en el segon nivell d'avís (<=min2 o >=max2)
- Veure en un gràfic les dades de les últimes 6 hores d'un sensor clicant en el valor d'un sensor de la pantalla anterior.
- Consultar log alarmes
- Validar alarmes
- Crear alarmes
- Gestionar comentaris d'alarmes
- Assignar una acció a una alarma
- Consultar històric dels valors d'un sensor
- Consultar en un gràfic el valor d'un sensor per un rang de dates
- Gestionar sensors, microcontroladors, etc.
- Veure mapa dels edificis
- Canviar contrasenya
- Desconnectar

I quines seran les funcionalitats d'un administrador, sempre i quan tingui connexió:

- Totes les funcionalitats d'un usuari.
- Crear usuari
- Assignar rol a usuari

- Veure registre de canvis (generats pels *triggers*)



5.2.1 Inicialitzar sessió

L'usuari introdueix el seu usuari i contrasenya, l'aplicació verifica si existeix i el redirigeix al menú que tingui assignat segons el seu rol.

Actors: Usuari amb connexió

Activació: Sempre que iniciem l'aplicació o després d'una desconnexió

Precondició: Estar registrat abans a l'aplicació per un usuari administrador. Tenir connexió 3g o Wifi.

Escenari Principal:

1. L'usuari inicialitza l'aplicació per primer cop.
2. El sistema li demanarà iniciar sessió.
3. L'usuari omple el formulari d'iniciar sessió.
4. El sistema el deixa accedir si ha sigut donat d'alta anteriorment, i té un rol associat.

5.2.2 Desconnectar

L'usuari desconnecta de l'aplicació

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Sortir"

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Sortir".
2. El sistema mostra el formulari de inicialitzar sessió.

5.2.3 Veure dades sensors per ubicació

L'usuari visualitza el llistat dels valors de tots els sensors d'una ubicació.

Actors: Usuari amb connexió

Activació: Al clicar la opció "Inici"

Precondició: Tenir la sessió inicialitzada.

Escenari Principal:

1. L'usuari clica la opció "Inici"
2. El sistema mostra les ubicacions disponibles
3. L'usuari selecciona una ubicació
4. El sistema mostra una pantalla amb tots els valors dels sensors d'una ubicació

5.2.4 Veure en un gràfic les dades de les últimes 6 hores d'un sensor

L'usuari visualitza en un gràfic les dades de les últimes sis hores d'un sensor.

Actors: Usuari amb connexió

Activació: Al clicar damunt un sensor en la pantalla "Veure dades sensor per ubicació"

Precondició: Tenir la sessió inicialitzada i estar en la pantalla anterior.

Escenari Principal:

1. L'usuari clica en un valor d'un sensor que es mostra en pantalla
2. El sistema mostra en un gràfic les dades de les últimes sis hores, si hi ha valors, del sensor seleccionat

5.2.5 Consultar registre d'alarmes

L'usuari consulta l'historial d'alarmes.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Administració Alarmes", i "Veure log alarmes"

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Administració Alarmes".
2. L'usuari clica l'opció "Veure log alarmes".
3. L'usuari selecciona si vol veure les alarmes verificades o no.
4. El sistema mostra l'historial de log d'alarmes verificades o no, segons hagi escollit l'usuari.

5.2.6 Validar alarmes

L'usuari valida una alarma no validada.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció d'editar una alarma no Verificada i la valida amb un comentari.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Administració Alarmes".
2. L'usuari clica l'opció "Veure log alarmes".
3. L'usuari selecciona alarmes no verificades.

4. L'usuari clica el llapis just al costat esquerra de l'alarma que vol validar.

5. L'usuari selecciona al camp "Verificada" el valor "Sí", i selecciona un comentari per validar-la.

6. El sistema verifica l'alarma i la passa a validada.

5.2.7 Crear alarmes

L'usuari crea una alarma per un sensor determinat.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Administració Alarmes" i després sobre "Veure/Crear alarmes" per crear una alarma.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Administració Alarmes".

2. L'usuari clica l'opció "Veure/Crear alarmes".

3. El sistema mostra una llista d'alarmes.

4. L'usuari pot modificar una alarma existent o bé crear una de nova amb el botó "Nova".

5.2.8 Gestionar comentaris d'alarmes

L'usuari gestiona els comentaris que podrem posar per validar les no validades.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Administració Alarmes" i després sobre "Comentaris d'alarmes" per gestionar els comentaris.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Administració Alarmes".

2. L'usuari clica l'opció "Comentaris d'alarmes".

3. El sistema mostra un llistat de comentaris que tenim per les alarmes.

4. L'usuari pot modificar un comentari existent o bé crear-ne un de nou amb el botó "Nou".

5.2.9 Assignar una acció a una alarma

L'usuari assigna una acció a una alarma.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Administració Alarmes" i després sobre "Accions" per poder assignar una acció a una alarma existent.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Administració Alarmes".

2. L'usuari clica l'opció "Accions".

3. El sistema mosta una llista d'accions per alarma que tenim registrats.

4. L'usuari pot modificar alguna acció existent o bé crear una de nova amb el botó "Nova".

5.2.10 Consultar històric valors d'un sensor

L'usuari consulta el històric de valors d'un sensor.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Veure dades" i fa la cerca del que li interessi.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Veure dades".
2. L'usuari clica l'opció "Cerca".
3. L'usuari selecciona el sensor del qual vol veure les dades.
4. L'usuari selecciona el període del qual vol veure les del sensor.
5. El sistema mostra el valor de les dades del sensor seleccionat pel període escollit, ordenat de més nou a més antic.

5.2.11 Consultar en un gràfic els valors d'un sensor per un rang de dates

L'usuari consulta el històric de valors d'un sensor en un gràfic per un rang de dates seleccionat.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Veure dades en un gràfic" i selecciona un sensor i un rang de dates.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Veure dades en un gràfic".
2. L'usuari selecciona un sensor.
3. L'usuari selecciona una data inici i una data fi.

4. El sistema mostra en un gràfic els valors de les dades del sensor escollit, si hi ha, per les dates seleccionades.

5.2.12 Gestionar valors aplicació

L'usuari gestiona els valors que el sistema utilitzarà per funcionar.

Actors: Usuari amb connexió

Activació: L'usuari clica sobre l'opció "Manteniment Administratiu" i escull què vol administrar.

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari clica la opció "Manteniment Administratiu".
2. L'usuari clica l'opció que vol gestionar.
3. L'usuari veu una llista de les dades actuals que hi ha del tipus escollit.
4. L'usuari pot crear o modificar una nova dada d'aquell tipus seleccionat.

5.2.13 Crear usuari

L'administrador crea un nou usuari que tindrà accés al sistema.

Actors: Administrador amb connexió

Activació: L'administrador clica sobre l'opció "Usuaris" i selecciona "Afegir/veure usuaris".

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'administrador clica l'opció "Usuaris".

2. L'administrador clica l'opció "Afegir/veure usuaris".

3. El sistema mostra una llista d'usuaris existents.

4. L'administrador escull entre editar un usuari actual o crear-ne un de nou mitjançant el botó "Nou".

5.2.14 Canviar contrasenya (usuari/administrador)

L'usuari/administrador canvia la seva contrasenya actual.

Actors: Usuari/administrador amb connexió

Activació: L'usuari/administrador clica sobre l'opció "Canviar contrasenya".

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari/administrador clica la opció "Canviar contrasenya".

2. L'usuari/administrador introdueix dos cops la nova contrasenya a utilitzar.

3. El sistema actualitza la contrasenya de l'usuari.

5.2.15 Veure mapes

L'usuari/administrador consulta el mapa d'un edifici.

Actors: Usuari/administrador amb connexió

Activació: L'usuari/administrador clica sobre l'opció "Veure mapes".

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'usuari/administrador clica la opció "Veure mapes".

2. L'usuari/administrador selecciona de quin edifici vol veure el mapa.

3. El sistema mostra el mapa per l'edifici seleccionat.

5.2.16 Assignar rol a usuari

L'administrador assigna un rol a un usuari nou. Només tenim dos rols possibles: usuari i administrador.

Actors: Administrador amb connexió

Activació: L'administrador clica sobre l'opció "Rols d'usuaris".

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'administrador clica la opció "Rols d'usuaris".

2. El sistema mostra una llista de rols per usuaris existents: usuari/administrador.

3. L'administrador tria entre editar el rol d'un usuari existent, o bé crear un rol per un usuari nou amb el botó "Nou".

5.2.17 Veure registre de canvis

L'administrador consulta el registre de canvis en taules, produïts pel llançament de triggers creats en la base de dades.

Actors: Administrador amb connexió

Activació: L'administrador clica sobre l'opció "Registre de canvis".

Precondició: Tenir una sessió activada.

Escenari Principal:

1. L'administrador clica la opció "Registre de canvis".

2.El sistema mostra un registre d'esdeveniments, ordenats per data ascendent, de tots els canvis que s'han produït ja siguin inserts, updates o deletes.

3. L'administrador pot filtrar per taula, tipus d'esdeveniment, per camp canviat seleccionant el botó "Cerca" i filtrant pel que desitgi.

6 Disseny de l'aplicació

6.1 Arquitectura lògica: disseny en capes

Per a desenvolupar l'aplicació, s'ha decidit utilitzar una arquitectura de programació per capes. Es tracta d'una arquitectura client-servidor en el que l'objectiu principal és la separació de la lògica de negoci (el que no veu l'usuari: comprovacions, modificacions, etc) de la lògica del disseny (el que veu l'usuari).

Capa de presentació: Anomenada també "Capa d'usuari". En les aplicacions web es coneix també amb el nom "Capa Web". És la que veu l'usuari. Presenta el sistema a aquest, li comunica la informació i captura la informació de l'usuari en un mínim de procés (realitza un filtrat previ per comprovar que no hi ha errors de format). També és coneguda com a interfície gràfica i ha de tenir la característica de ser "amigable" (fàcil d'entendre i d'utilitzar) per l'usuari. Aquesta capa es comunica únicament amb la capa de negoci.

Capa de negoci: Anomenada també "Capa Aplicativa". És on resideixen els programes que s'executen, es reben les peticions de l'usuari i s'envien les respostes després de processar-les. Se li diu capa de negoci perquè és aquí on s'estableixen totes les regles que s'han de complir. Aquesta capa es comunica amb la de presentació per rebre les sol·licituds i presentar els resultats, i amb la capa de dades, per sol·licitar al gestor de base de dades guardar o recuperar dades d'ell.

Capa de dades: Coneguda com a "Capa de Base de Dades". És on resideixen les dades i l'encarregada d'accedir a elles. Està formada per un o més gestors de bases de dades que realitzen tot el emmagatzemament de dades, reben sol·licituds d'emmagatzemament o recuperació de informació des de la capa de negoci (20).

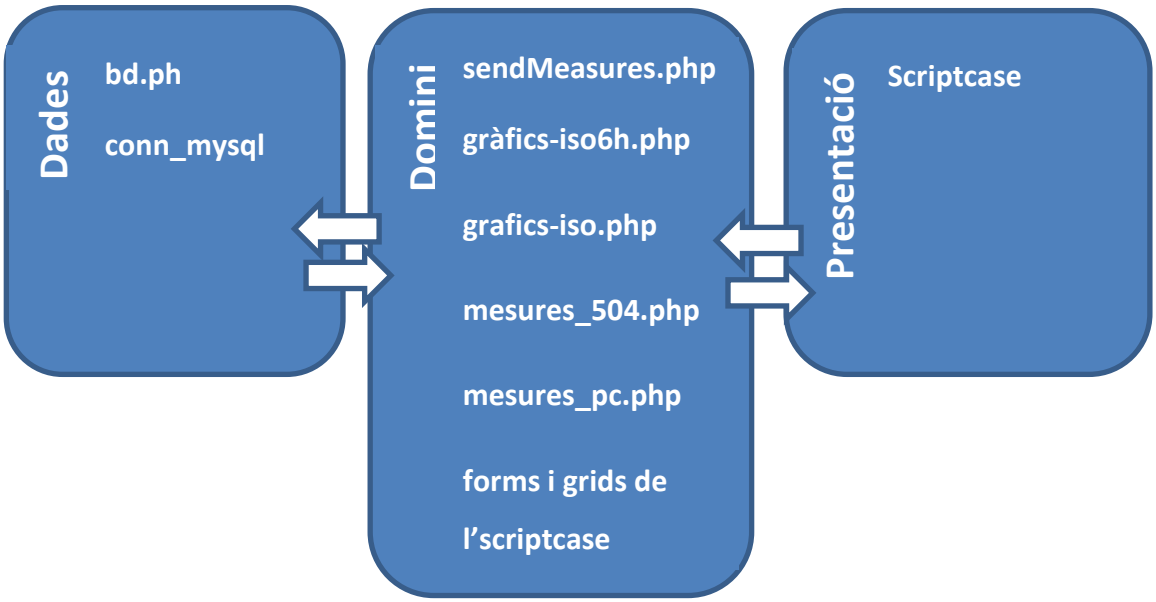
En el projecte es pretén utilitzar aquesta arquitectura dividint el codi en tres parts

Presentació (UIL): Conté els fitxers *php* amb la interfície d'usuari.

Dades (DAL): Conté tot el que és referent a crides a base de dades i connexió amb el *webervices*.

Negoci (BLL): Conté totes les funcions que es necessiten per a emplenar de dades les pantalles, enviar dades, comprovacions...

A continuació un esquema per a entendre el concepte:



Per exemple. Si l'usuari es troba en el menú de *l'Scriptcase* "Veure dades en un gràfic", de la capa de presentació, per poder visualitzar les dades, es cridarà a "gràfics-iso.php" de la capa de domini, que farà la petició a la capa de dades mitjançant la connexió a la base de dades "bd.php", i les mostrarà per pantalla.

Un altre exemple seria si un usuari volgués donar d'alta un comentari per les alarmes. Aniria a la opció de *l'Scriptcase* "Manteniment d'alarmes" → "Comentaris d'alarmes" de la capa de presentació, que aniria a buscar un formulari intern de l'Scripcase a la capa de domini "form_alarmes", que sol·licitaria a la capa de dades mitjançant la connexió "Conn_mysql" la introducció d'aquell comentari a la base de dades.

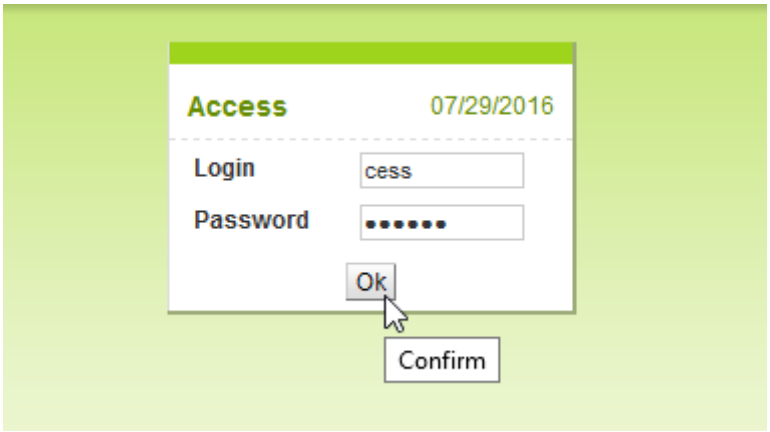
6.2 Disseny de la capa de presentació

Per a aquesta capa, s'utilitza el software *Scriptcase*. Com anteriorment hem explicat, aquest software utilitza el llenguatge PHP.

S'utilitzaran crides *a webservices* externs, també programats utilitzant codi PHP, per tal de veure els gràfics dels valors dels sensors.

A continuació el mapa de navegació de l'aplicació, per tal de veure com seran les pantalles, quines funcionalitats tindran els seus components i com és la transició entre les pantalles. Aquí no s'inclou el disseny *CSS* de les pantalles, simplement l'estructura i funcionalitats.

6.2.1 Pantalla d'inicialitzar sessió



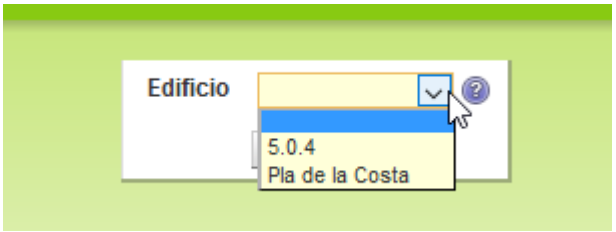
És tracta de la primera pantalla amb la que l'usuari interactuarà. Aquí introduirà el seu usuari i contrasenya per a poder inicialitzar sessió. Cal que aquest nom d'usuari i contrasenya ja existeixin, en cas negatiu, l'usuari hauria de demanar a un usuari administrador, que el donés d'alta.

Al clicar el botó "Ok", si les dades són correctes i es disposa de connexió a internet (3g o WiFi), l'usuari accedirà al menú principal.

6.2.2 Pantalla del Menú Principal

En la pantalla principal trobem 9 botons que al clicar-los ens portaran a les diferents funcionalitats:

- Inici: Du a l'usuari a l'inici del menú principal, on pot escollir un edifici del qual vol veure les dades dels seus sensors.



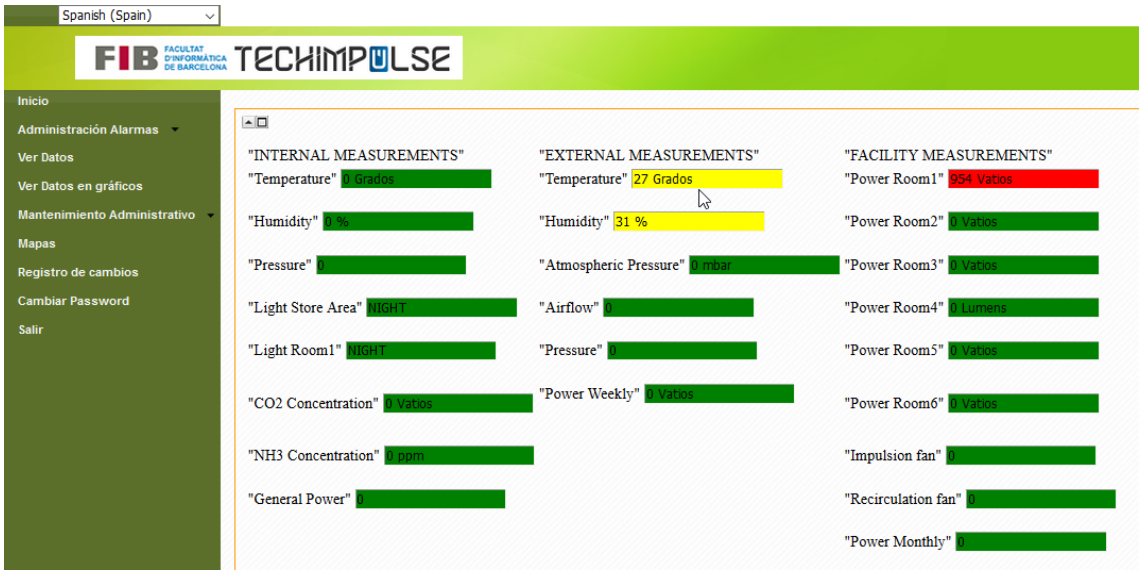
- Administració d'Alarmes: Dóna a l'usuari varies opcions. Veure el log, crear alarmes, gestionar els comentaris i assignar accions a alarmes.

- Veure dades: Mostra a l'usuari una llista dels valors de tots els sensors ordenats per data de més nou a més antic. L'usuari pot fer cerques per filtrar si visualitzar només les dades d'un sensor i en quina data.
- Veure dades en gràfic: Du a l'usuari a un selector per triar el sensor i la data inici i fi, per mostrar-li les dades d'un sensor en un gràfic en aquell període de temps.

- Manteniment administratiu: Du a l'usuari a administrar valors bàsics de la base de dades com: sensors, microcontroladors, accions, tipus de sensors, tipus de mesures, etc.
- Mapes: Mostra a l'usuari el mapa de cada edifici.
- Registre de canvis: Només visible per l'usuari administrador. Li mostra el registre de canvis (insercions, modificacions i eliminacions) produïts pels triggers.
- Canviar password: Deixa a l'usuari modificar la seva contrasenya.
- Sortir: Desconnecta al usuari de l'aplicació.

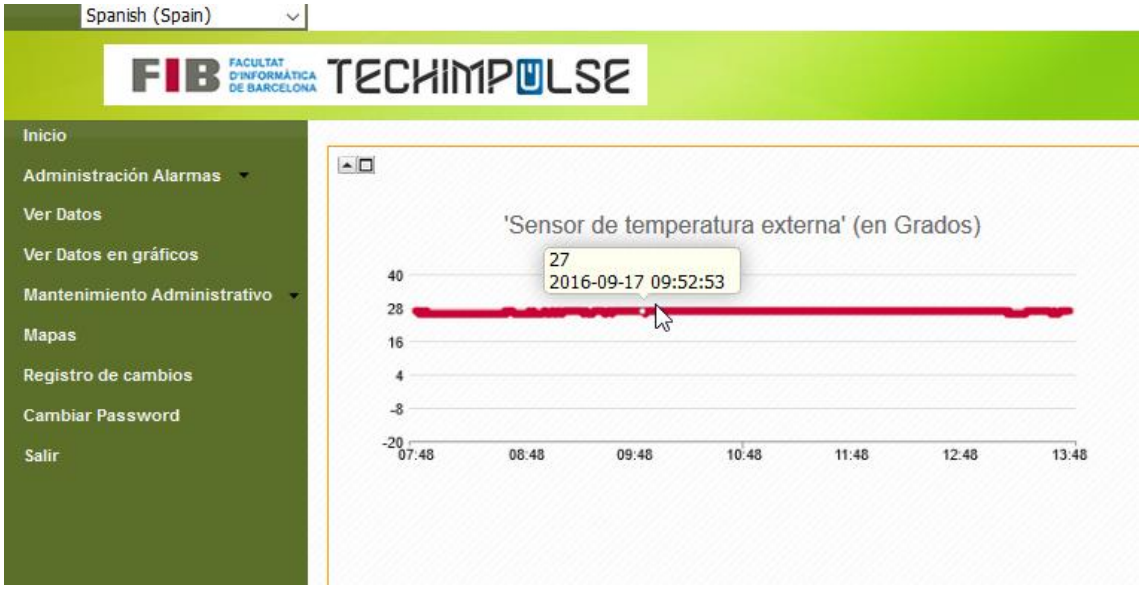
6.2.3 Pantalla Inici

Un cop l'usuari ha escollit un edifici, mostra els valors de tots els sensors que tenim, en colors per a que puguem veure fàcilment si la mesura és correcta o no.



En aquesta pantalla, l'usuari podrà interactuar de forma tal que, quan premi un valor d'un sensor el sistema el redirigirà a una pantalla on li mostrarà en un gràfic les dades d'aquell sensor en les últimes 6 hores.

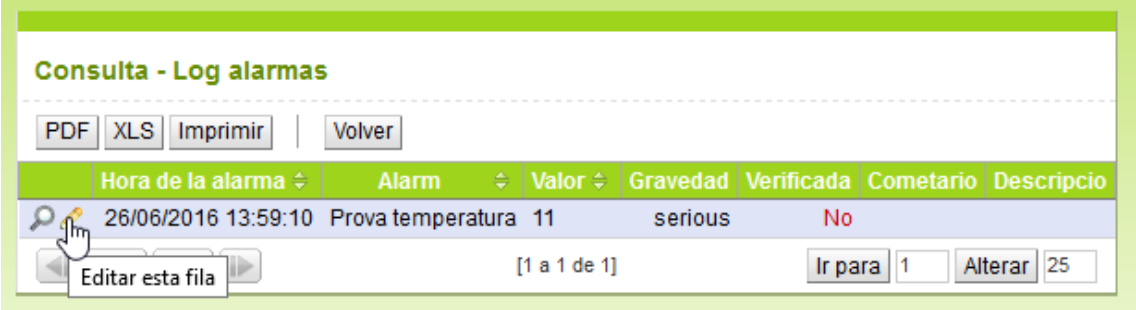
Per exemple en aquesta última imatge ens posem sobre el sensor de temperatura externa, i cliquem en el seu valor. Ens mostra el següent:



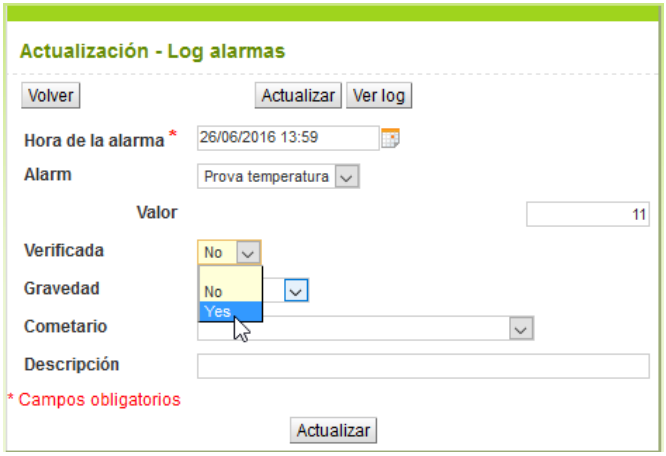
6.2.4 Pantalla Administració alarmes: "Veure log"

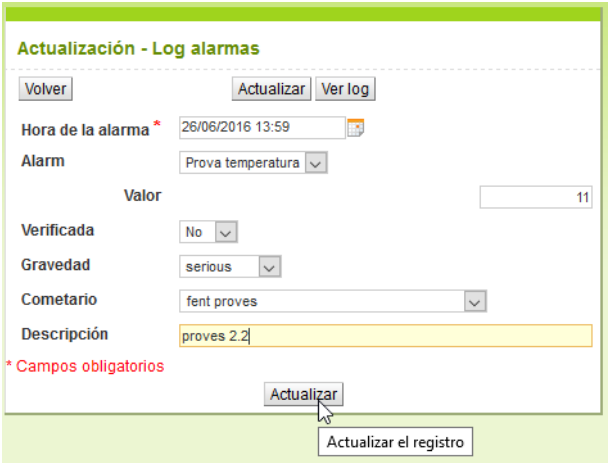
Primer de tot ens trobarem amb el filtre de alarmes. Verificades o no. Escollim no verificades.

El sistema ens mostrarà les alarmes no verificades, on podrem editar-la per verificar-la.



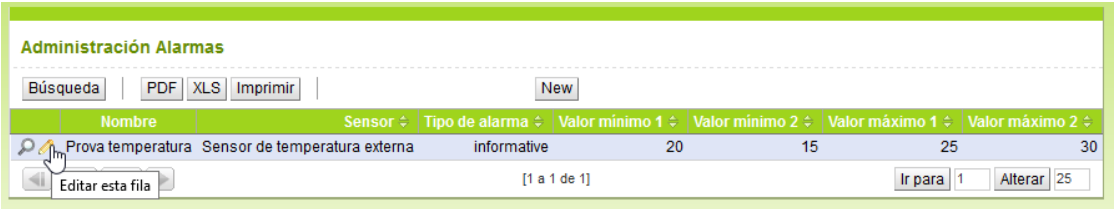
Modifiquem el seu estat a "Verificada" i li assignem un comentari, i si volem, una descripció.





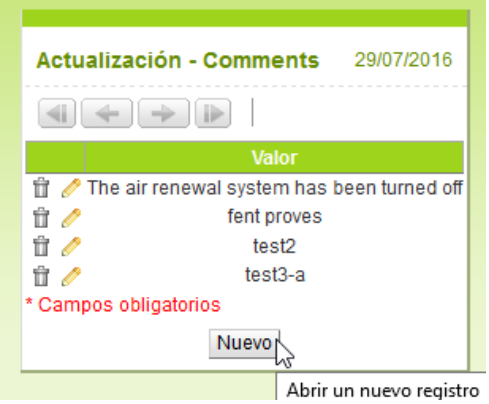
6.2.5 Pantalla Administració alarmes: "Veure/Crear alarmes"

Aquesta pantalla ens mostra totes les alarmes que tenim creades. Les podem modificar, eliminar o bé crear noves alarmes.



6.2.6 Pantalla Administració alarmes: "Comentaris d'alarmes"

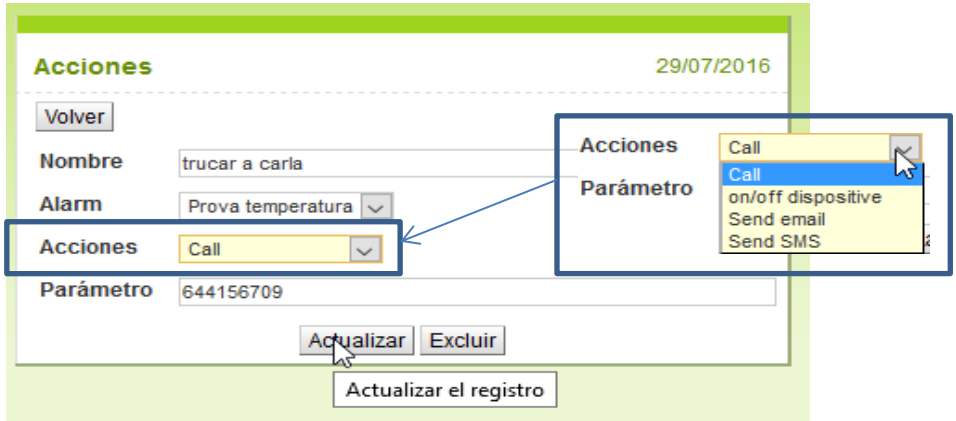
En aquesta pantalla podem gestionar els comentaris que podrem seleccionar en la validació de les alarmes. Podem crear, editar i eliminar comentaris.



6.2.7 Pantalla Administració alarmes: "Accions"

En aquesta pantalla és on podem assignar una acció a una alarma.

Veiem que ja tenim creada una acció per a la alarma "Prova de temperatura", vista anteriorment, que consisteix en trucar a un número, però tenim altres tipus d'accions. L'únic que tindrem que canviar, és el paràmetre. Si és enviar un correu, li haurem de posar una direcció de correu, si és trucar o enviar missatge, un número de telèfon (mòbil en el segon cas), i en el cas de activar/desactivar dispositiu de moment no caldrà res ja que ho tindrem fixat a un *led*.



6.2.8 Pantalla veure dades

En aquesta pantalla l'usuari podrà veure un registre de totes les mesures dels sensors, ordenades per data de més nova a més antiga.

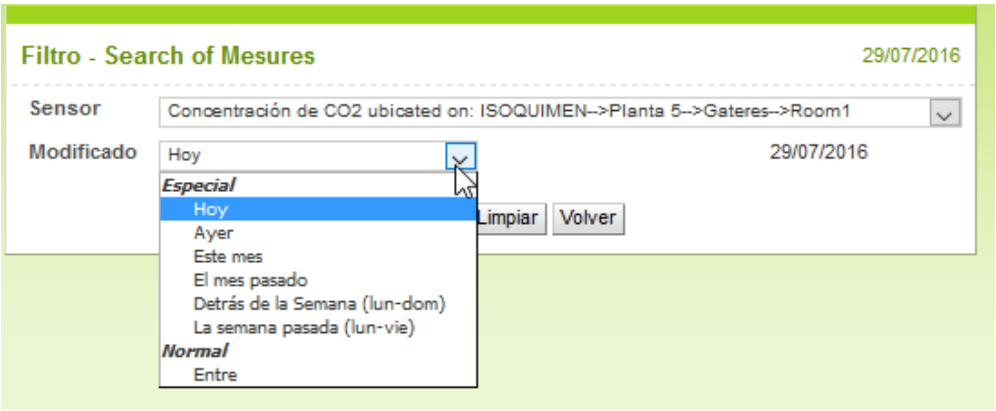
Log Mesures

Búsqueda | PDF | XLS | Imprimir

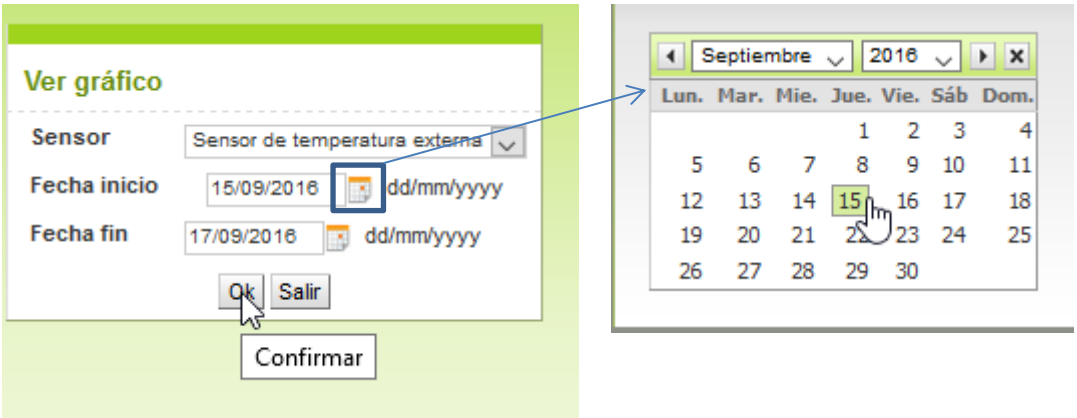
Microcontrolador	Sensor	Descripción	Valor	Unidades	Modificado
AABBCCDDEE01	Sensor de Humedad externa	Sensor de Humedad externa	50,00	%	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de temperatura externa	Sensor de temperatura externa	11,00	Grados	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Luz Room1	Sensor de Luz Room1	-0,05	Lumens	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de potencia Room1	Sensor de potencia Room1	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de potencia Room2	Sensor de potencia Room2	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de temperatura interna	Sensor de temperatura interna	0,00	Grados	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Humedad interna	Sensor de Humedad interna	0,00	%	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de potencia Room3	Sensor de potencia Room3	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Presión	Presión	0,00	mbar	08/07/2016 18:22:12
AABBCCDDEE01	Presión Atmosférica	Presión Atmosférica	0,00	mbar	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Luz Room2	Sensor de Luz Room2		Lumens	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Luz Room3	Sensor de Luz Room3		Lumens	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Potencia Room4	Sensor de Potencia Room4	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de potencia Room5	Sensor de potencia Room5	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Sensor de potencia Room6	Sensor de potencia Room6	0,00	Vatios	08/07/2016 18:22:12
AABBCCDDEE01	Concentración de CO2	Concentración de CO2	0,00	ppm	08/07/2016 18:22:12
AABBCCDDEE01	Concentración de NH3	Concentración de NH3	0,00	ppm	08/07/2016 18:22:12
AABBCCDDEE01	sensor porta room1	sensor porta room1	0,00		08/07/2016 18:22:12
AABBCCDDEE01	Sensor de Humedad externa	Sensor de Humedad externa	50,00	%	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de temperatura externa	Sensor de temperatura externa	11,00	Grados	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de Luz Room1	Sensor de Luz Room1	-0,05	Lumens	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de potencia Room1	Sensor de potencia Room1	0,00	Vatios	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de potencia Room2	Sensor de potencia Room2	0,00	Vatios	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de temperatura interna	Sensor de temperatura interna	0,00	Grados	08/07/2016 18:22:10
AABBCCDDEE01	Sensor de Humedad interna	Sensor de Humedad interna	0,00	%	08/07/2016 18:22:10

[1 a 25 de 816] Ir para 1 Alterar 25

Com que podem arribar a tenir molts sensors i moltes mesures, tenim un filtre de cerca, en el qual li podem especificar el sensor i el període del qual volem veure les mesures.

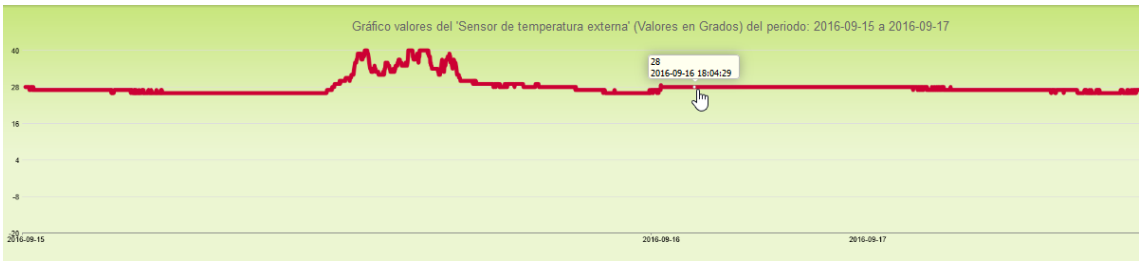


6.2.9 Pantalla veure dades en gràfic



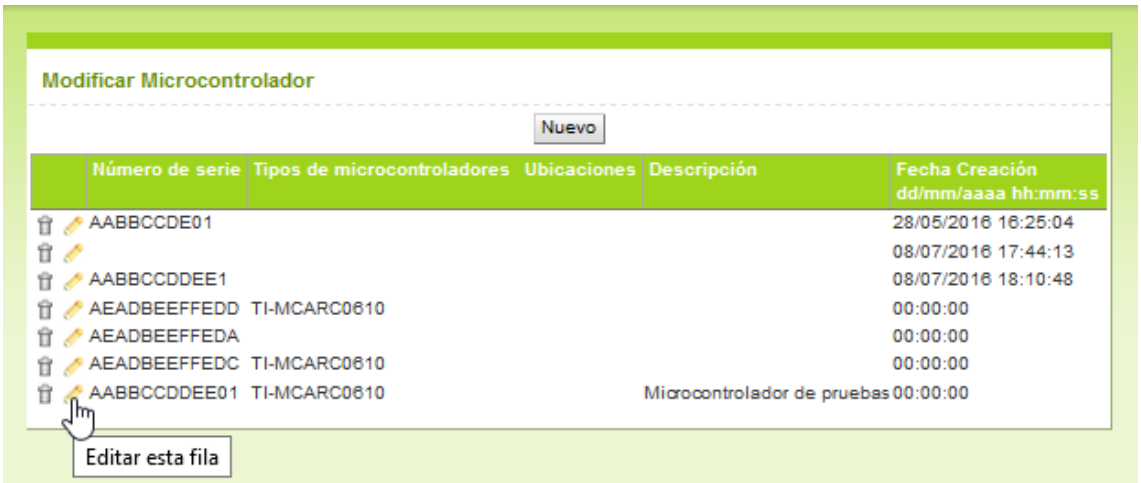
En aquesta pantalla l'usuari haurà d'escollir un sensor del qual vol veure les dades en un gràfic, una data inici i una data fi.

El sistema mostrarà les dades que tingui, si n'hi ha.



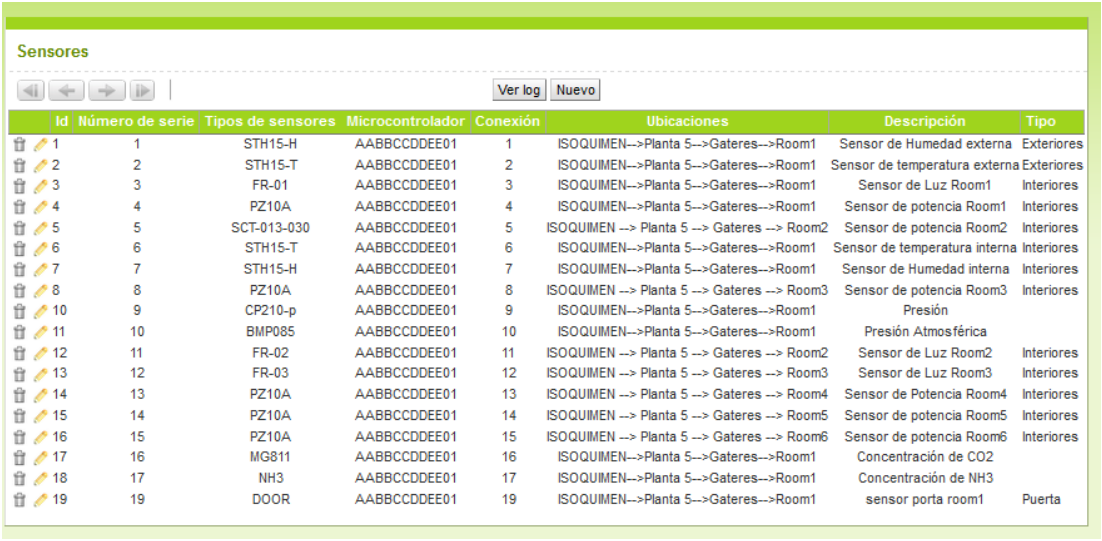
6.2.10 Pantalla manteniment administratius: "Microcontroladors"

En aquesta pantalla l'usuari podrà administrar els microcontroladors que té.



6.2.11 Pantalla manteniment administratius: "Sensors"

En aquesta pantalla l'usuari podrà administrar els sensors que té.

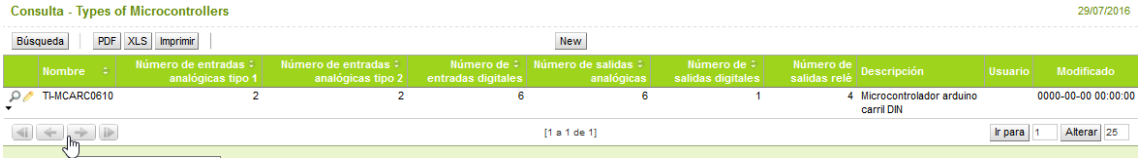


Id	Número de serie	Tipos de sensores	Microcontrolador	Conexión	Ubicaciones	Descripción	Tipo
1	1	STH15-H	AABBCCDDEE01	1	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de Humedad externa	Exteriores
2	2	STH15-T	AABBCCDDEE01	2	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de temperatura externa	Exteriores
3	3	FR-01	AABBCCDDEE01	3	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de Luz Room1	Interiores
4	4	PZ10A	AABBCCDDEE01	4	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de potencia Room1	Interiores
5	5	SCT-013-030	AABBCCDDEE01	5	ISOQUIMEN --> Planta 5 --> Gateres --> Room2	Sensor de potencia Room2	Interiores
6	6	STH15-T	AABBCCDDEE01	6	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de temperatura interna	Interiores
7	7	STH15-H	AABBCCDDEE01	7	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Sensor de Humedad interna	Interiores
8	8	PZ10A	AABBCCDDEE01	8	ISOQUIMEN --> Planta 5 --> Gateres --> Room3	Sensor de potencia Room3	Interiores
10	9	CP210-p	AABBCCDDEE01	9	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Presión	
11	10	BMP085	AABBCCDDEE01	10	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Presión Atmosférica	
12	11	FR-02	AABBCCDDEE01	11	ISOQUIMEN --> Planta 5 --> Gateres --> Room2	Sensor de Luz Room2	Interiores
13	12	FR-03	AABBCCDDEE01	12	ISOQUIMEN --> Planta 5 --> Gateres --> Room3	Sensor de Luz Room3	Interiores
14	13	PZ10A	AABBCCDDEE01	13	ISOQUIMEN --> Planta 5 --> Gateres --> Room4	Sensor de Potencia Room4	Interiores
15	14	PZ10A	AABBCCDDEE01	14	ISOQUIMEN --> Planta 5 --> Gateres --> Room5	Sensor de potencia Room5	Interiores
16	15	PZ10A	AABBCCDDEE01	15	ISOQUIMEN --> Planta 5 --> Gateres --> Room6	Sensor de potencia Room6	Interiores
17	16	MG811	AABBCCDDEE01	16	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Concentración de CO2	
18	17	NH3	AABBCCDDEE01	17	ISOQUIMEN-->Planta 5-->Gateres-->Room1	Concentración de NH3	
19	19	DOOR	AABBCCDDEE01	19	ISOQUIMEN-->Planta 5-->Gateres-->Room1	sensor porta room1	Puerta

Donar d'alta un sensor nou mitjançant el botó "Nou". Haurà d'associar el nou sensor a un microcontrolador existent, i a un pin o connexió.

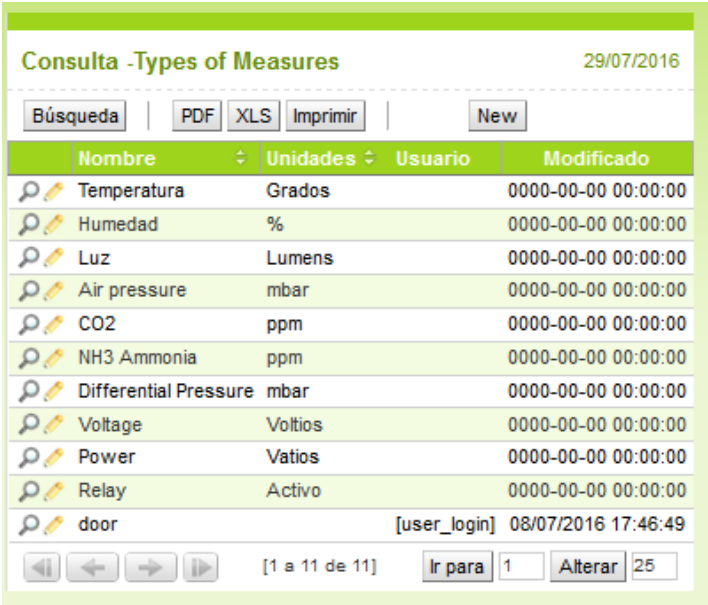
6.2.12 Pantalla manteniment administratius: "Tipus de microcontroladors"

En aquesta pantalla l'usuari podrà gestionar els tipus de microcontroladors.



6.2.13 Pantalla manteniment administratius: "Tipus de mesura"

En aquesta pantalla l'usuari podrà administrar els tipus de mesura dels sensors.



6.2.14 Pantalla manteniment administratius: "Tipus de sensors"

En aquesta pantalla l'usuari podrà administrar els tipus de sensors. Li assignarem un tipus de mesura, un tipus de sortida i una fórmula de càlcul.

Tipos de sensores 29/07/2016

Búsqueda | PDF | XLS | Imprimir | Nuevo

Nombre	Tipos de medida	Tipo de salida	Fórmula de cálculo	Tipos de sensor	Usuario	Modificado
STH15-T	Temperatura	Digital				0000-00-00 00:00:00
STH15-H	Humedad	Digital				0000-00-00 00:00:00
BMP085	Air pressure	Analog				0000-00-00 00:00:00
MG811	CO2	Analog				0000-00-00 00:00:00
FR-01	Luz	Analog	(((2500/($y*5/1023))-500)/10000)			0000-00-00 00:00:00
CP210-T	Temperatura	Analog				0000-00-00 00:00:00
CP210-p	Differential Pressure	Analog	$x*(100/1023)			0000-00-00 00:00:00
VC	Voltage	Analog				0000-00-00 00:00:00
PZ10A	Power	Analog				0000-00-00 00:00:00
RL10A	Relay	Digital 0 or 1				0000-00-00 00:00:00
TH110-T	Temperatura	Analog	($x*(100/1023))-20			0000-00-00 00:00:00
TH110H	Humedad	Analog	(((($x*10/1023)/0.111111111)+5)			0000-00-00 00:00:00
SHT10	NH3 Ammonia	Analog	($X*220)			0000-00-00 00:00:00
SCT-013-030	Power		($x*210)			0000-00-00 00:00:00
CP210-c	door		(sqrt($y/0.575))*52.8			0000-00-00 00:00:00
FR-02	Luz		50/(((5-($x*5/1023))/($x*5/1023))			0000-00-00 00:00:00
NH3	NH3 Ammonia		($x*5/1023)			0000-00-00 00:00:00
TH210-T	Temperatura		($x*100/2046)			0000-00-00 00:00:00
TH210-H	Humedad		($x*100/1023)			0000-00-00 00:00:00
CP210-b0-pa	Differential Pressure		($x/10)			0000-00-00 00:00:00
CP210-b0-m3	door					0000-00-00 00:00:00
C310B0-T	Temperatura		($x*100/2046)			0000-00-00 00:00:00
C310B0-H	Humedad		($x*100/1023)			0000-00-00 00:00:00
DOOR	door	Digital 0 or 1			Puerta [user_login]	08/07/2016 17:47:02
FR-03	Luz		50/(3*(((5-($x*5/1023))/($x*5/1023)))			0000-00-00 00:00:00

[1 a 25 de 25] Ir para 1 Alterar 25

6.2.15 Pantalla manteniment administratius: "Tipus d'accions"

En aquesta pantalla l'usuari podrà gestionar els tipus d'accions, tot i que s'ha de tocar a nivell de codi què fa cada acció.

Consulta - Types of Actions 29/07/2016

Búsqueda | PDF | XLS | Imprimir | New

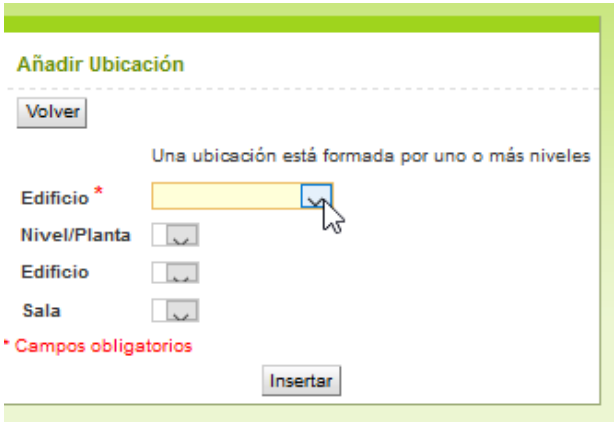
Acciones	Cometario	Usuario	Modificado
Send email	carla.balcells@esadecreapolis.com	[user_login]	28/05/2016 12:08:34
Send SMS	644156709		0000-00-00 00:00:00
Call	truca a un número		0000-00-00 00:00:00
on/off dispositive	Canvia l'estat on/off un dispositiu		0000-00-00 00:00:00

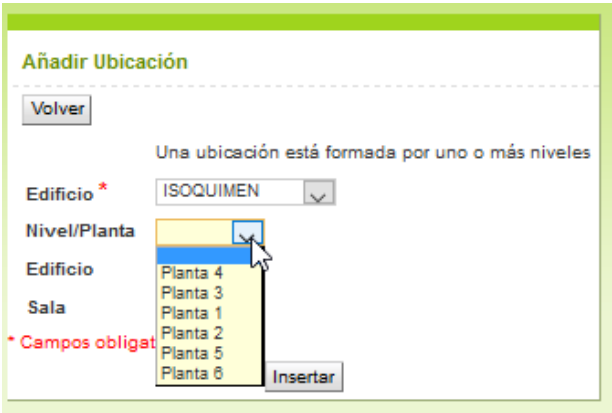
[1 a 4 de 4] Ir para 1 Alterar 25

6.2.16 Pantalla manteniment administratius: "Ubicacions"

En aquesta pantalla l'usuari podrà definir ubicacions, definint primer els nivells que tindrà cada una.

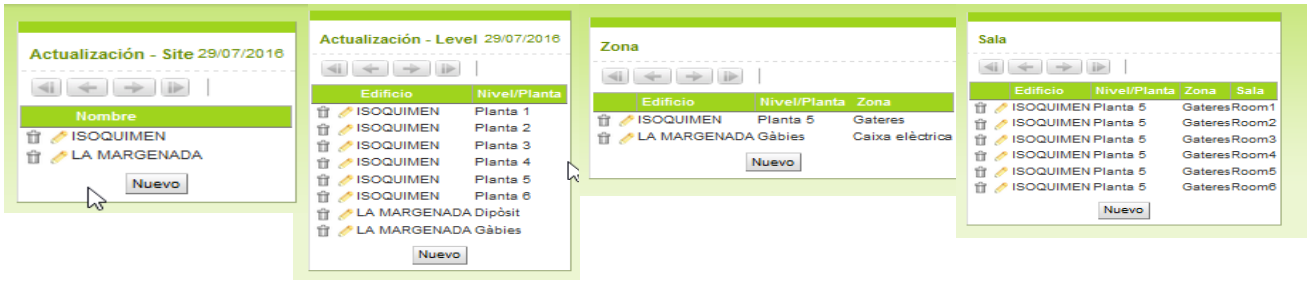
Podem crear noves ubicacions.





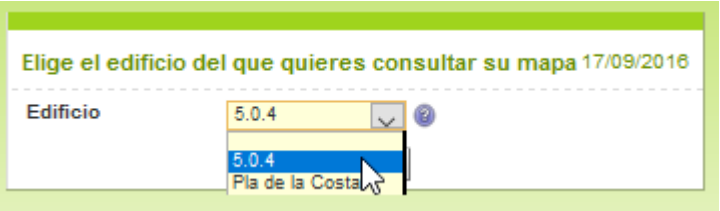
6.2.17 Pantalla manteniment administratius: definir nivells: "Edifici", "Nivell/Planta", "Zona" i "Sala"

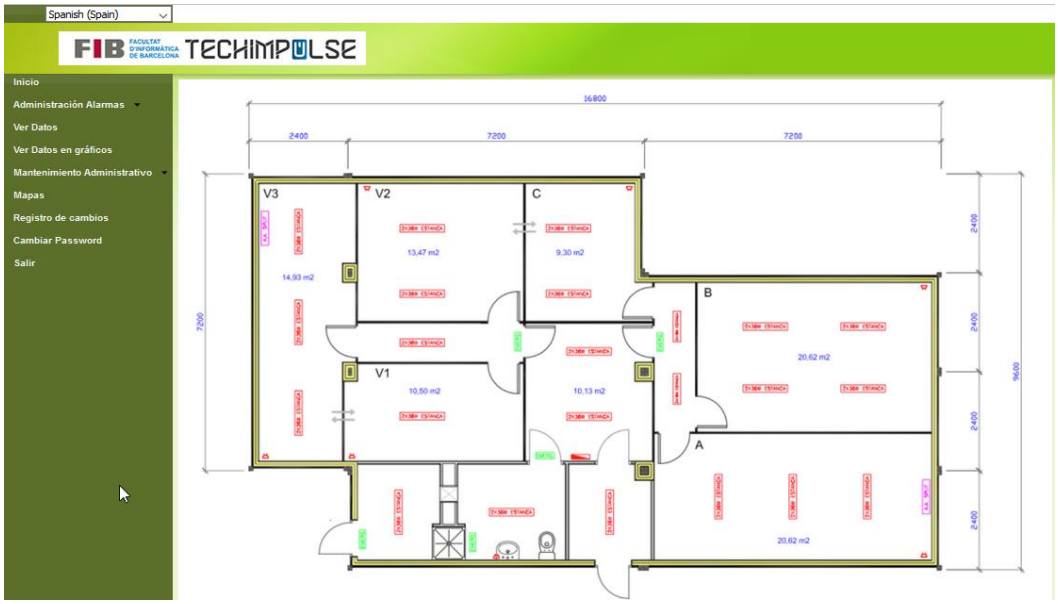
En aquestes pantalles l'usuari crearà a nivell de edifici, planta, zona i sala, una lògica per poder crear una ubicació amb elles.



6.2.18 Pantalla "Veure mapes"

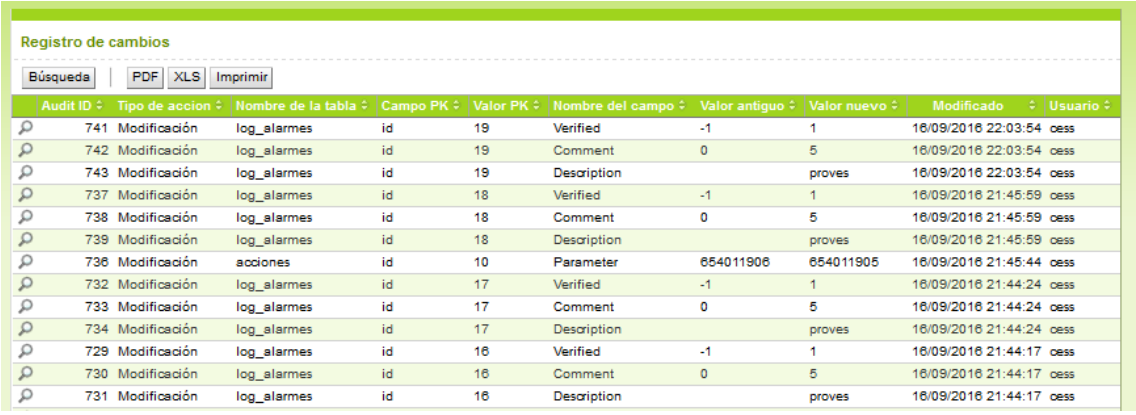
En aquesta pantalla, un usuari pot seleccionar el edifici del qual vol veure el seu mapa, i el sistema li mostrarà.





6.2.19 Pantalla "Veure registre canvis"

En aquesta pantalla, un administrador pot veure els canvis produïts pels triggers, al modificar-se, inserta o eliminar qualsevol cosa de qualsevol taula.

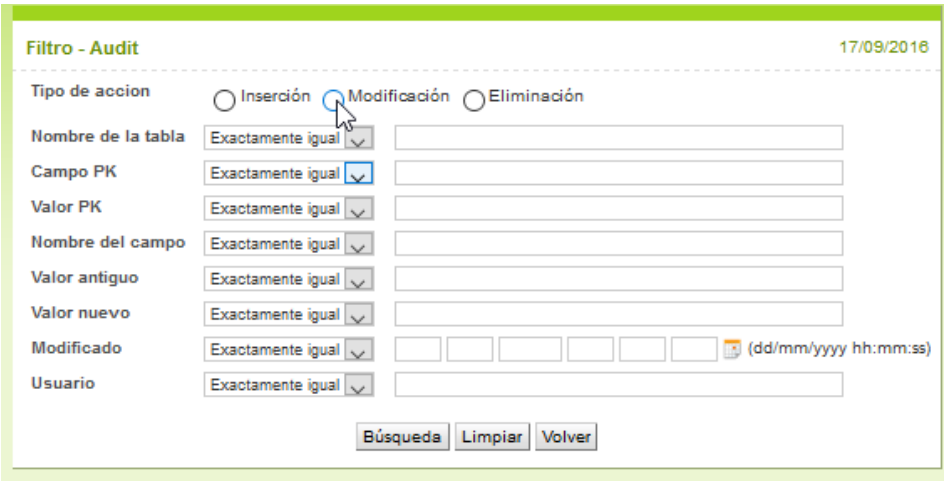


Registro de cambios

Búsqueda | PDF XLS Imprimir

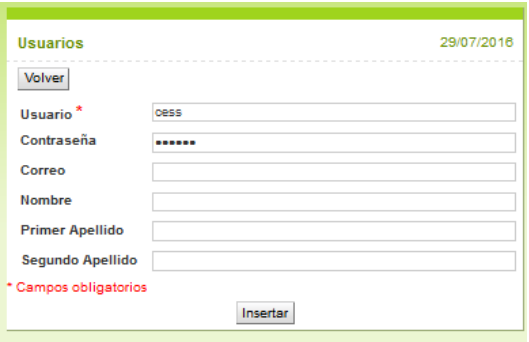
Audit ID	Tipo de accion	Nombre de la tabla	Campo PK	Valor PK	Nombre del campo	Valor antiguo	Valor nuevo	Modificado	Usuario
741	Modificación	log_alarmes	id	19	Verified	-1	1	16/09/2016 22:03:54	cess
742	Modificación	log_alarmes	id	19	Comment	0	5	16/09/2016 22:03:54	cess
743	Modificación	log_alarmes	id	19	Description		proves	16/09/2016 22:03:54	cess
737	Modificación	log_alarmes	id	18	Verified	-1	1	16/09/2016 21:45:59	cess
738	Modificación	log_alarmes	id	18	Comment	0	5	16/09/2016 21:45:59	cess
739	Modificación	log_alarmes	id	18	Description		proves	16/09/2016 21:45:59	cess
736	Modificación	acciones	id	10	Parameter	654011906	654011905	16/09/2016 21:45:44	cess
732	Modificación	log_alarmes	id	17	Verified	-1	1	16/09/2016 21:44:24	cess
733	Modificación	log_alarmes	id	17	Comment	0	5	16/09/2016 21:44:24	cess
734	Modificación	log_alarmes	id	17	Description		proves	16/09/2016 21:44:24	cess
729	Modificación	log_alarmes	id	16	Verified	-1	1	16/09/2016 21:44:17	cess
730	Modificación	log_alarmes	id	16	Comment	0	5	16/09/2016 21:44:17	cess
731	Modificación	log_alarmes	id	16	Description		proves	16/09/2016 21:44:17	cess

Si vol veure els registres d'una data en concret, els inserts/updates/deletes, el que ha fet un usuari en concret, etc, pot filtrar amb el botó "Cerca" i escollir el que vol veure.



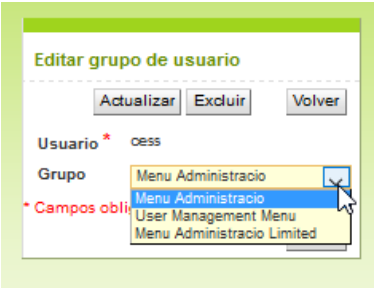
6.2.20 Pantalla manteniment administratius: "Usuaris" → "Crear Usuari"

En aquesta pantalla, un administrador pot crear nou usuaris o modificar-ne un de ja creat.

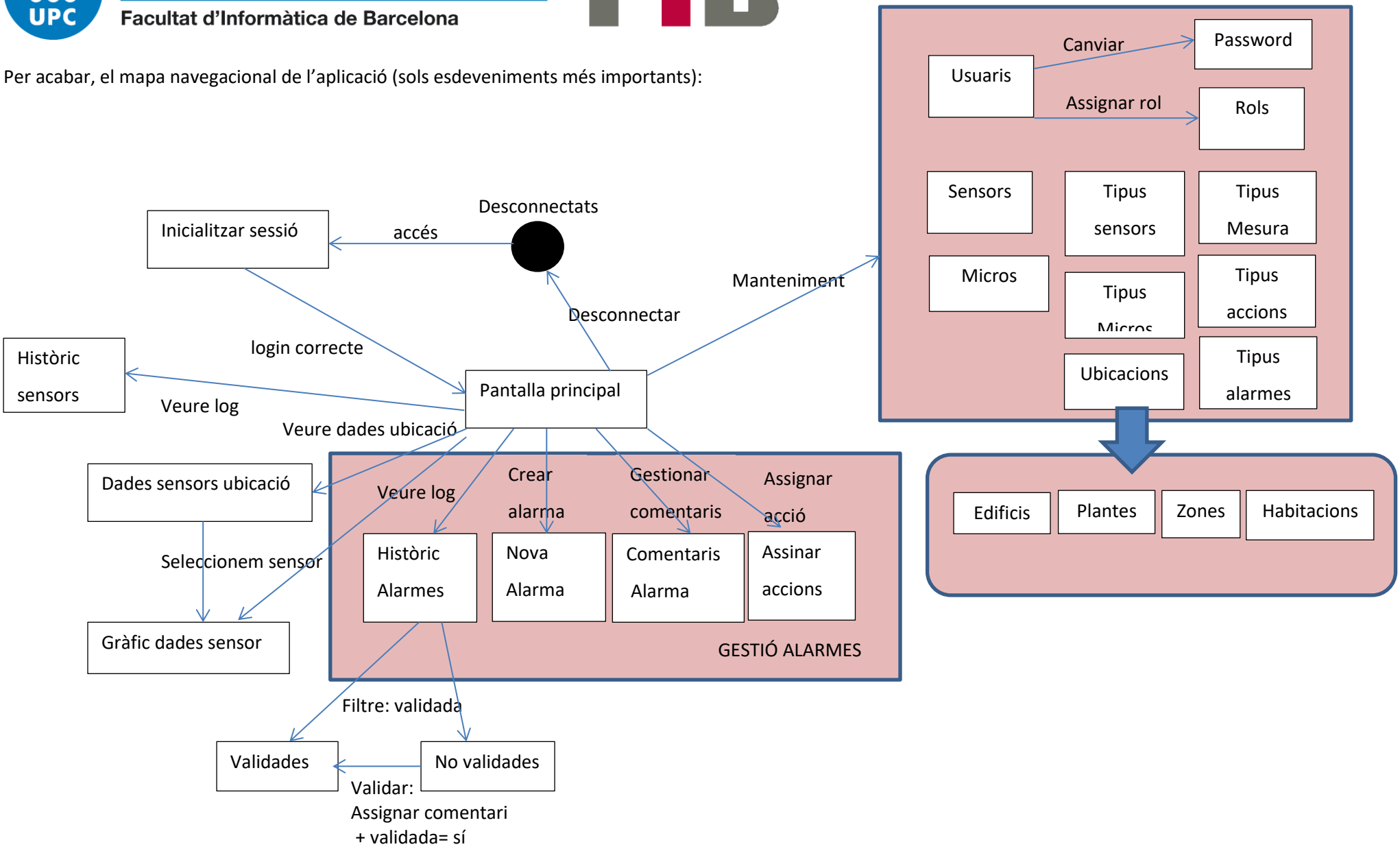


6.2.21 Pantalla manteniment administratius: "Usuaris" → "Veure Grups d'usuaris"

En aquesta pantalla, un administrador podrà assignar un rol a un usuari existent.



Per acabar, el mapa navegacional de l'aplicació (sols esdeveniments més importants):



6.3 Disseny de la capa de domini

Aquesta capa conté tota la lògica del projecte, per tant és aquí on:

- Es realitzen tot tipus de comprovacions.
- Conté les funcions per a les transicions entre pantalles.
- Es recuperen valors de la capa de dades per a assignar-los a elements de la capa de presentació.
- Es recuperen les dades de la capa de presentació per a poder-los afegir a la base de dades o servidor des de la capa de dades.

Al tenir la capa de presentació feta amb el generador de codi PHP *Scriptcase*, moltes coses les gestiona ell mateix internament.

Nosaltres haurem de:

- Definir el *select* SQL de la taula a fer el formulari o *grid*.
- Escollir quins camps mostrarem.
- Definir de quin tipus és cada camp i el nom que tindrà cara a l'usuari final. *Scriptcase* té diferents tipus de camps com E-mail, codi de barres, text, text de múltiples línies, decimals i molts més. Cada tipus de dada es pot restringir encara més, com un decimal fins a quantes dècimes tindrà, un camp de text quin tipus de caràcters deixarem introduir (telèfons, NIF's...), codi de barres de quin tipus (EAN13, EAN15...), etc.
- Claus primàries: per a cada formulari tenim que definir quins camps seran les claus primàries. Normalment això es defineix a nivell de base de dades, i al crear el formulari *Scriptcase* les agafa automàticament, però pot ser que vulguem modificar-ho.
- Camps obligatoris: per a cada formulari tenim que definir quins camps seran obligatoris. Normalment això es defineix a nivell de base de dades, i al crear el formulari *Scriptcase* les agafa automàticament, però pot ser que vulguem modificar algun.
- Comprovacions en els esdeveniments que té en els formularis de creació/modificació de nous tipus de dades com:
    - "onAplicationInit": accions que volem realitzar tot just al iniciar.

- "onValidate": quan li donem a qualsevol botó, les comprovacions/accions que té que fer per aquell registre.
- "onBeforeInsert": comprovacions/accions a fer abans de crear un nou registre.
- "onBeforeUpdate": comprovacions/accions a fer abans d'actualitzar un registre.
- "onBeforeDelete": comprovacions/accions a fer abans d'eliminar un registre. Pot ser que al eliminar un sensor, volem eliminar tot el seu històric de valors.
- "onValidateSuccess": comprovacions/accions si hem passat amb èxit el "onValidate".

Aquestes serien unes, també tindríem altres com: onAfterInsert, onAfterUpdate, onAfterDelete, onRefresh, onNavigate, etc.

A part de *Scriptcase*, tenim *webservices* externs:

- sendMeasures.php: Aquest *webservice* és el que rep les mesures del *Arduino*. Es crida en la programació del microcontrolador, cada 50 segons. Realitza les següents tasques:
  - Comprovacions de paràmetres entrants: mac del microcontrolador, valors s1→s21.
  - Comprovació de connexió amb la base de dades.
  - Inserció dels valors rebuts per microcontrolador en la hora actual.
  - Comprovació dels valors per sensor.
  - Comprovació d'alarmes per sensor.
  - Comprovació de si existeix ja una alarma d'aquell nivell. Sinó, la crea.
  - Comprovació d'accions per alarmes.
  - Execució d'accions.

- grafics-iso.php: Aquest *webservice* ens mostra en un gràfic els valors d'un sensor en un rang de dates. Es crida des del backoffice, quan anem a Veure dades d'un sensor en un gràfic.
  - Comprovacions de paràmetres entrants: id del sensor, data inici i data fi.

- Comprovació de connexió amb la base de dades.
- Recollida de dades per aquell sensor en aquell període de dates.
- Preparació de l'eix x (dates entre els dos valors que ens han passat), l'eix y (y mínim i y màxim), els valors del sensor i les etiquetes que es mostraran.
- Creació del gràfic si hi ha dates.
- Mostrar gràfic resultant.

- grafics-iso6h.php: : Aquest *webservice* ens mostra en un gràfic els valors d'un sensor durant les últimes sis hores. Es crida des de la pantalla proporcionada pels *webservice* "mesures_504.php" o "mesures_pc.php", al clicar sobre un dels valors dels sensors actius.
  - Comprovacions de paràmetres entrants: id sensor.
  - Comprovació de connexió amb la base de dades.
  - Recollida de dades per aquell sensor en les últimes sis hores.
  - Preparació de l'eix x (últimes sis hores), l'eix y (y mínim i y màxim), els valors del sensor i les etiquetes que es mostraran.
  - Creació del gràfic si hi ha dates.
  - Mostrar gràfic resultant.

- mesures_504.php: Aquest *webservice* mostra en una pantalla tots els valors dels sensors d'aquella ubicació. Es crida des del *backoffice*, quan escollim veure les dades dels sensors de l'edifici "504".
  - Comprovació de connexió amb la base de dades.
  - Comprovació de dades per tots els sensors de la ubicació 504.
  - Comprovació de si cada sensor té alguna alarma, i si en té pintar l'etiqueta del valor que li correspongui: verd si el valor és correcte, groc si el valor està entre mínim1 i mínim2 o bé màxim1 i màxim2, i vermell si el valor és >=màxim2 o bé <=mínim2.
  - Preparació de cada etiqueta del valor del sensor per a que enllaci amb el seu gràfic de les últimes sis hores corresponent.
  - Preparació de cada etiqueta del valor del sensor, concatenant-lo amb el seu tipus de dades corresponent.

- Comprovació de si el tipus de sensor és de llum, mostrar "DAY" en cas de que el valor sigui superior a 3.5 *lumens*, o bé "NIGHT" en cas de que sigui inferior.
- Mostra totes les dades finals.

- mesures_pc.php: Aquest *webservice* mostra en una pantalla tots els valors dels sensors d'aquella ubicació. Es crida des del *backoffice*, quan escollim veure les dades dels sensors de l'edifici "pc".
  - Comprovació de connexió amb la base de dades.
  - Comprovació de dades per tots els sensors de la ubicació 504.
  - Comprovació de si cada sensor té alguna alarma, i si en té pintar l'etiqueta del valor que li correspongui: verd si el valor és correcte, groc si el valor està entre mínim1 i mínim2 o bé màxim1 i màxim2, i vermell si el valor és >=màxim2 o bé <=mínim2.
  - Preparació de cada etiqueta del valor del sensor per a que enllaci amb el seu gràfic de les últimes sis hores corresponent.
  - Preparació de cada etiqueta del valor del sensor, concatenant-lo amb el seu tipus de dades corresponent.
  - Comprovació de si el tipus de sensor és de llum, mostrar "DAY" en cas de que el valor sigui superior a 3.5 *lumens*, o bé "NIGHT" en cas de que sigui inferior.
  - Mostra totes les dades finals.

6.3.1 Comprovació de paràmetres entrants

El paràmetres que ens entrin als *webservices*, tenen que ser vàlids.

Es comprova mitjançant GET, que tots els paràmetres siguin correctes.

6.3.2 Comprovació de connexió amb la base de dades

Es realitza de dos formes diferents.

En *Scriptcase* amb la connexió que hem creat al crear el projecte: "conn_mysql", on hem creat la connexió amb la IP del servidor, usuari, contrasenya i seleccionant la base de dades. La resta ho fa sol.

En els nostres *webservices* amb un fitxer "bd.php", que és una classe creada per nosaltres que realitza la connexió, consulta i modificació de dades.

6.4 Disseny de la capa de dades

En aquesta capa s'hi gestiona tot el que està relacionat amb dades:

- Accés a la base de dades del servidor des de *Scriptcase*.
- Accés a la base de dades del servidor des dels *webservices*.

6.4.1 Base de dades del servidor

S'utilitza una *bbdd* externa, en un altre servidor, de tipus *mysql*. Aquesta base de dades és accessible mitjançant un *phpmyadmin* que hi hem instal·lat.

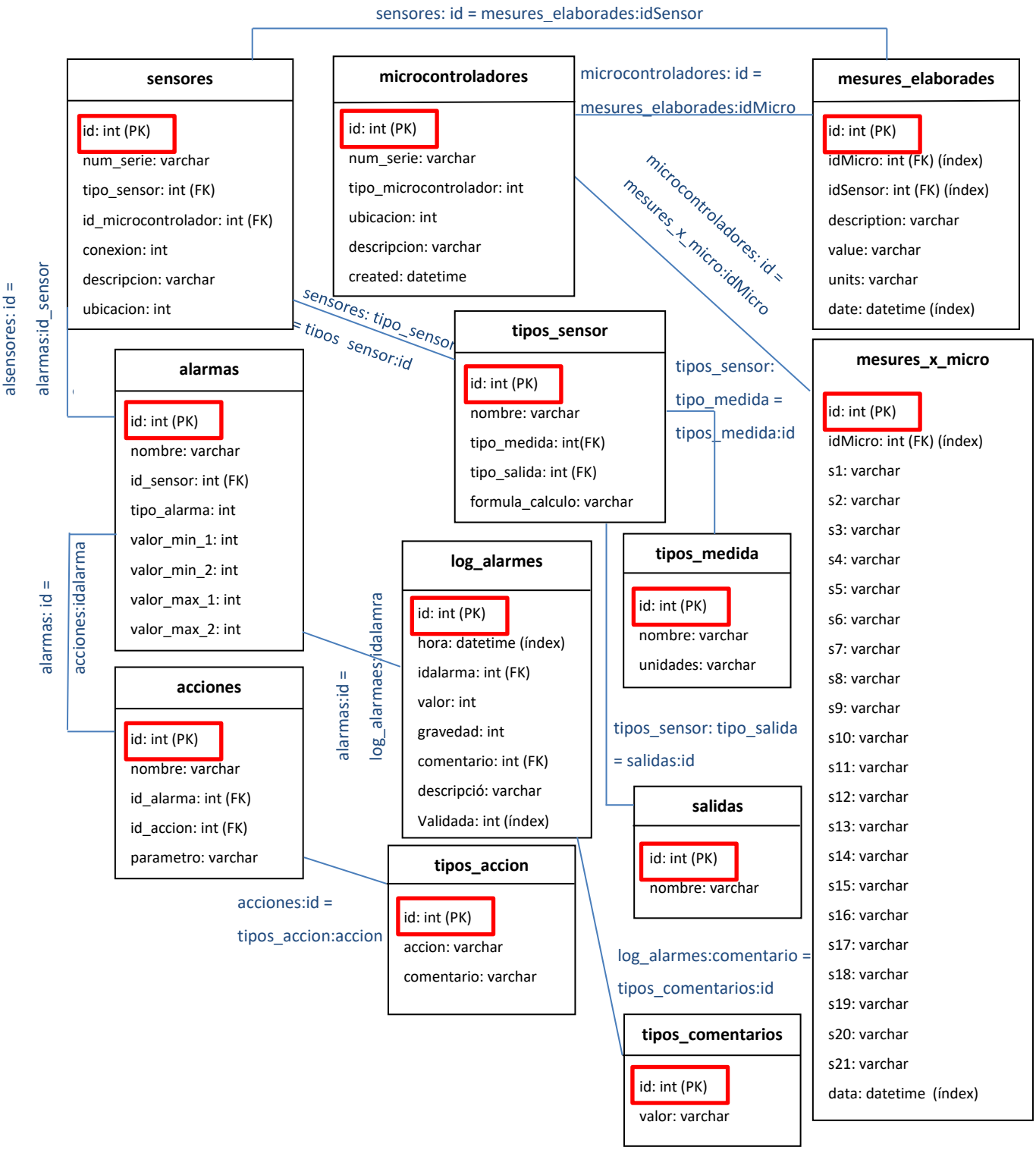
Scriptcase accedeix a la base de dades mitjançant la connexió que hem definit al crear el projecte: "conn_mysql", on hem creat la connexió definint la direcció/nom del servidor on es troba la *bbdd*, i un usuari i contrasenya que hem creat des de la *bbdd* prèviament. Un cop són validades, ens mostra un llistat de les bases de dades a les quals aquell usuari té accés, seleccionem la base de dades a la qual ens volem enllaçar i la resta ho fa sol.

Els nostres *webservices* hi accedeixen mitjançant un fitxer "bd.php", que és una classe creada per nosaltres que realitza la connexió, consulta i modificació de dades.

Els serveis webs que es connectaran a la base de dades seran els comentats anteriorment:

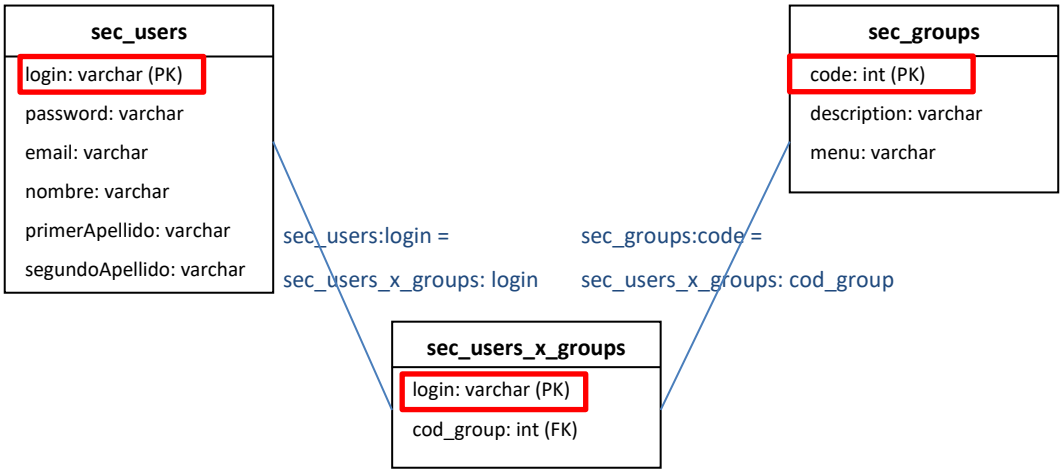
- sendMeasures.php
- grafics-iso.php
- grafics-iso6h.php
- mesures_504.php
- mesures_pc.php

A continuació veurem el disseny de la base de dades:



Com hem vist, els índexs són pocs, ja que les úniques cerques complexes que fem són les de mesures de valors dels microcontroladors, i normalment les filtrem per data. Tenim també un índex en les alarmes, pel camp Validada, ja que també volem filtrar ràpidament per aquest valor.

Per altra banda, tenim la part d'usuaris, que no es relaciona amb les altres dades:



7 Implementació

7.1 Entorn de desenvolupament

7.1.1 Instal·lació entorn Scriptcase

En el entorn de desenvolupament primer de tot necessitem tenir *Scriptcase* instal·lat. Nosaltres treballarem amb la versió 5.0 d'aquest software, que ens podem baixar des de l'enllaç que veurem al programari. També necessitem tenir una llicència per poder treballar-hi.

A més a més per a que funcioni, necessitarem de:

- Servidor *web* amb suport per *php: Apache*
- PHP
- Una versió estàndard de base de dades: Nosaltres treballarem amb *Mysql*

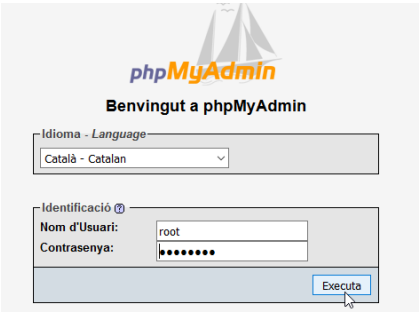
A més a més instal·larem *Phpmyadmin* per poder accedir i crear l'estructura de la base de dades més ràpidament.

Per instal·lar-lo, seguim els passos de instal·lació que veurem també, al programari.

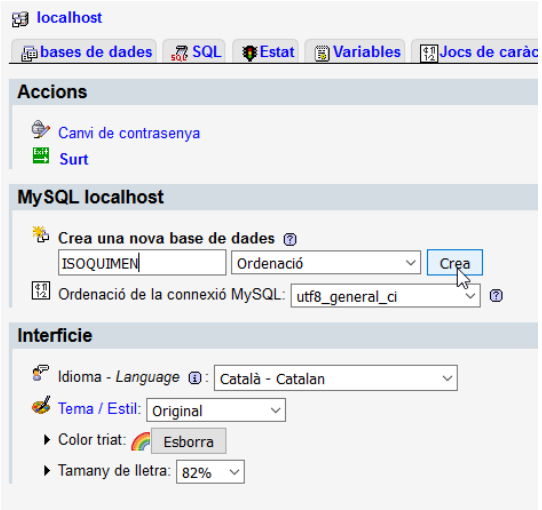
7.1.2 Base de dades

Un cop instal·lat, hauríem de crear un projecte, però primer hem de crear la base de dades amb la que l'associarem.

Com hem instal·lat el *phpmyadmin*, podem accedir via web amb la "ip/phpmadmin" al gestor de base de dades i començar a crear-la.



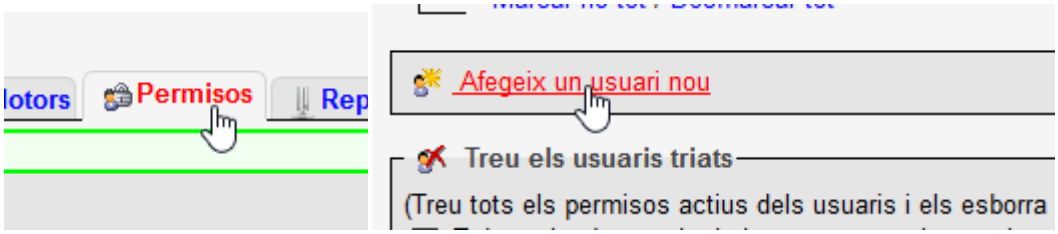
Primer de tot crearem la base de dades.

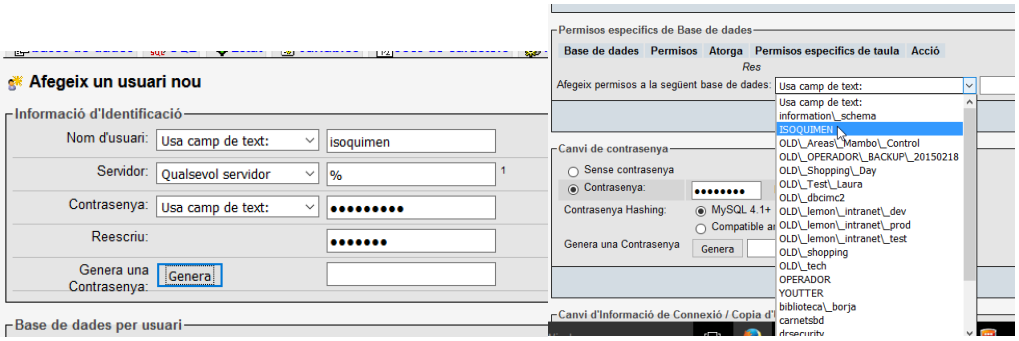


Per poder connectar-nos a la *bbdd*, tant els *webservices* com *l'Scriptcase* necessitaran d'un usuari amb accés a ella.

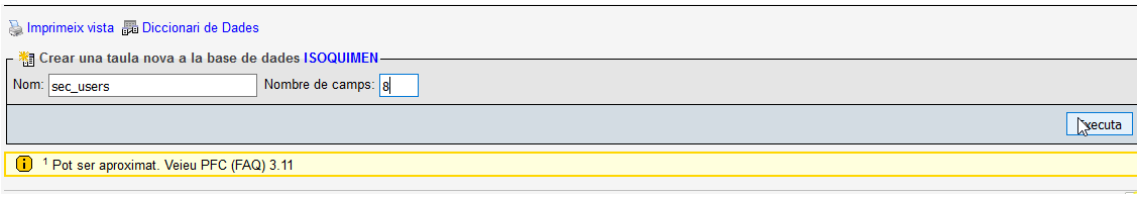
El creem des de bdcimc.esadecreapolis.com

Sense seleccionar cap *bbdd*, anem a la pestanya "Usuarios" → "Agregar usuario" i creem el nostre usuari donant-li accés des de qualsevol servidor: % amb privilegis complets en la *bbdd* que hem creat.





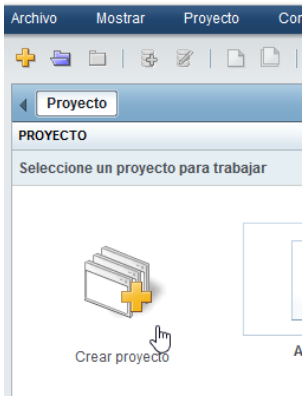
Un cop fet això, ja podem començar a crear les taules.



7.1.3 Creació projecte i connexió amb base de dades

Un cop hem arribat aquí, ja tenim tota l'estructura de la base de dades amb la que treballarem. Ara ens toca començar a generar els formularis i grids que veurà l'usuari en la capa de presentació.

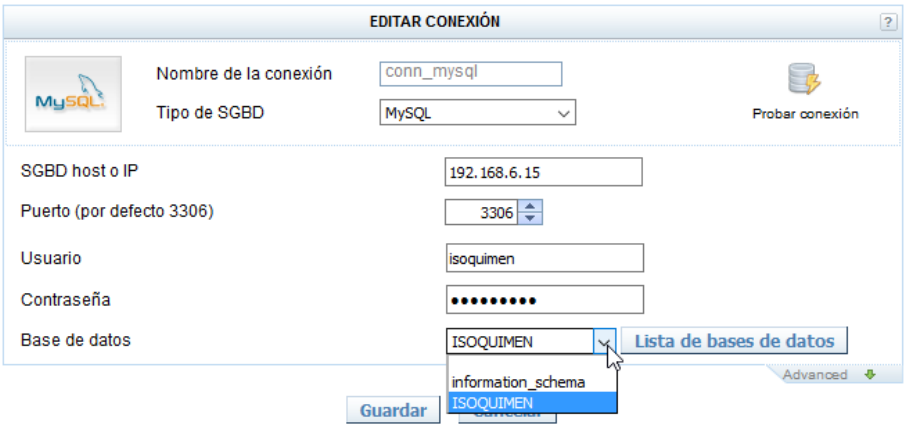
Anem al *Scriptcase* instal·lat, que podrem accedir-hi mitjançant un navegador qualsevol, i creem un projecte nou.



Ens demana detalls bàsics com els idiomes que suportarà, idioma predeterminat dins d'ells, temes (plantilles *CSS* que du *Scriptcase*) que utilitzarem per la visualització i tema predeterminat dins d'ells, quins usuaris hi tindran accés i quins d'ells seran administradors del projecte.

Un cop fet tot això, ens demanarà de crear una connexió del projecte amb la base de dades. Nosaltres seleccionarem una connexió de tipus "Mysql".

I allà posem les dades de l'usuari que hem creat al *phpmyadmin*:



7.1.4 Instal·lació software microcontrolador

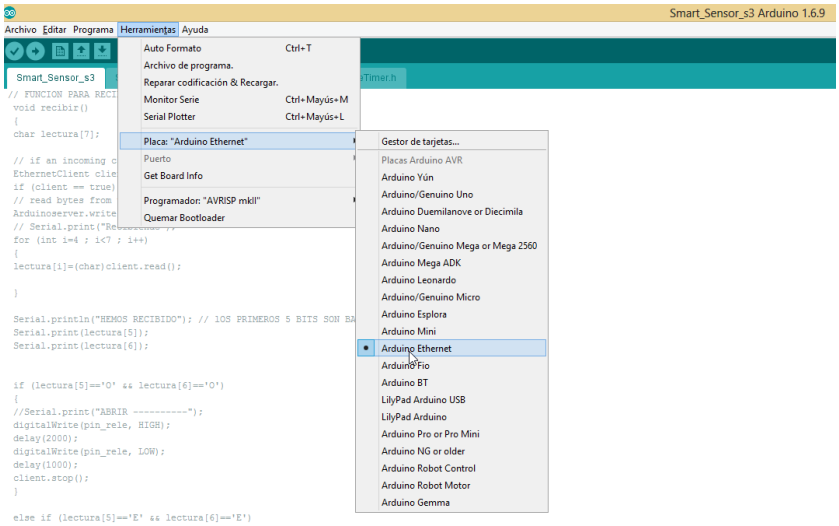
Com utilitzarem el nostre portàtil per programar el microcontrolador, també necessitarem tenir els *drivers* i el software per programar-lo. Utilitzarem *Arduino IDE*, la última versió en el moment: la 1.6.9. Junt a aquest, també necessitarem instal·lar Java Runtime Enviroment (J2RE).

Una vegada tinguem *l'Arduino* IDE instal·lat, necessitarem baixar-nos e importar en el nostre codi les llibreries que haguem d'utilitzar:

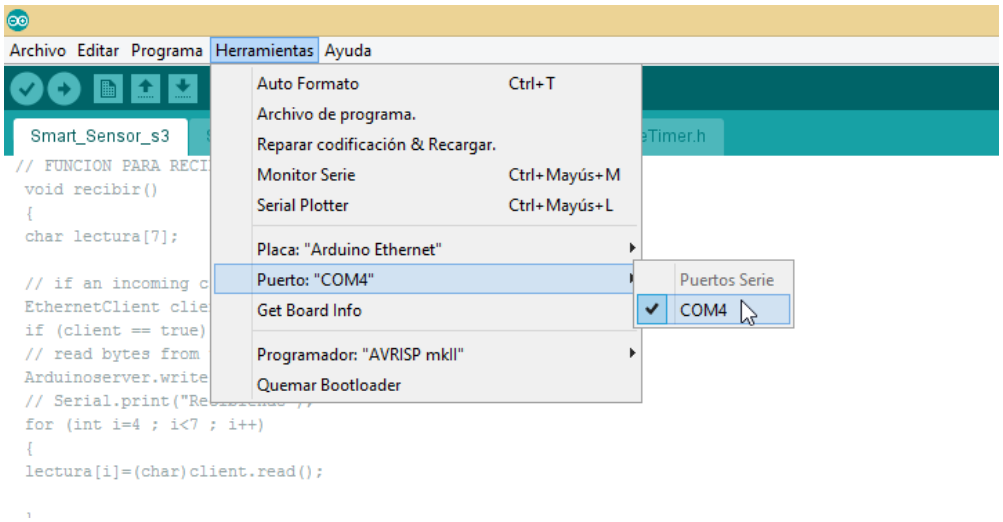
- "Ethernet": per poder configurar el port ethernet de *l'Arduino*, i poder enviar les mesures al *webservice* del servidor extern.
- "SimpleTimer": la utilitzarem per poder programar l'enviament de mesures cada 50 segons.

- "SHT1X": Al ser el *SHT15* un sensor analògic, *Arduino* no pot llegir e seu valor directament. Necessita d'una llibreria específica d'aquest sensor d'humitat i temperatura. Amb aquestes llibreries podem crear instàncies del sensor i utilitzar les funcions de lectura varies de les que disposa. En aquest cas aquest sensor disposa de lectura de temperatura en Faradays i Graus, i lectura de humitat (21).

En el *Arduino IDE* li indicarem que treballarem amb un *Arduino* Ethernet:



I quan el tinguem connectat haurem de buscar el port COM pel qual ens el detecti:



7.1.5 Webservices

El desenvolupament dels *webservices* el realitzarem des del nostre equip, per ftp directament al servidor de desenvolupament.

Per fer-ho, al servidor necessitarem:

- PHP
- Mysql client

En el nostre equip necessitarem:

- Notepad++ (modificarem el codi php directament per ftp)

7.2 Detalls de la implementació

7.2.1 Material necessari

Abans de posar-nos a tocar codi, necessitarem fer el muntatge dels sensors en la placa.

Necessitem el material següent:

- Placa *Arduino*
- Cable USB per programar-lo

- Cables mascle-mascle, mascle-femella o conversos mascle-femella per fer les connexions entre sensors, placa i *Protoboard*

- Sensor SHT15
- Sensor de llum
- Sensor de porta magnètica
- Placa *Protoboard*
- Led
- Relés
- Soldador
- Estany

Primer de tot mirarem si tenim suficients ports per posar tots els sensors. *Arduino* disposa de 14 pins digitals, 6 analògics, 3 pins de GND (*ground*) i 1 de voltatge. Si no tenim suficients amb aquests, sempre ens podem ajudar de la *Protoboard* per duplicar el valor de terra o el de voltatge.

7.2.2 Muntatge dels sensors

7.2.2.1 Sensor de temperatura i humitat

El sensor SHT15, és una placa sense pins de connexió, fet pel qual l'haurem de soldar en una placa.

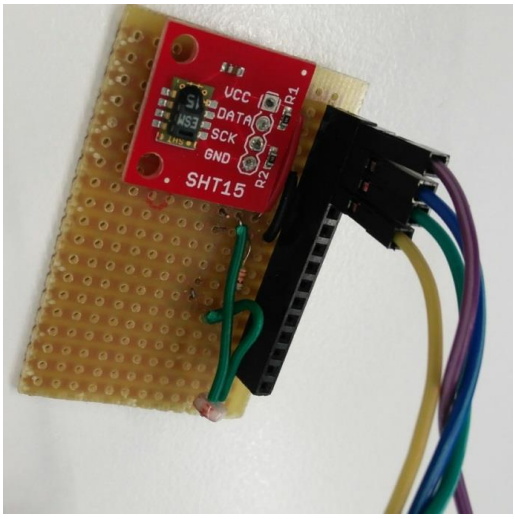
Per soldar el sensor necessitarem: una placa, estany, soldador i el sensor.

El soldarem de tal forma que cada connexió quedi en línia amb un pin.

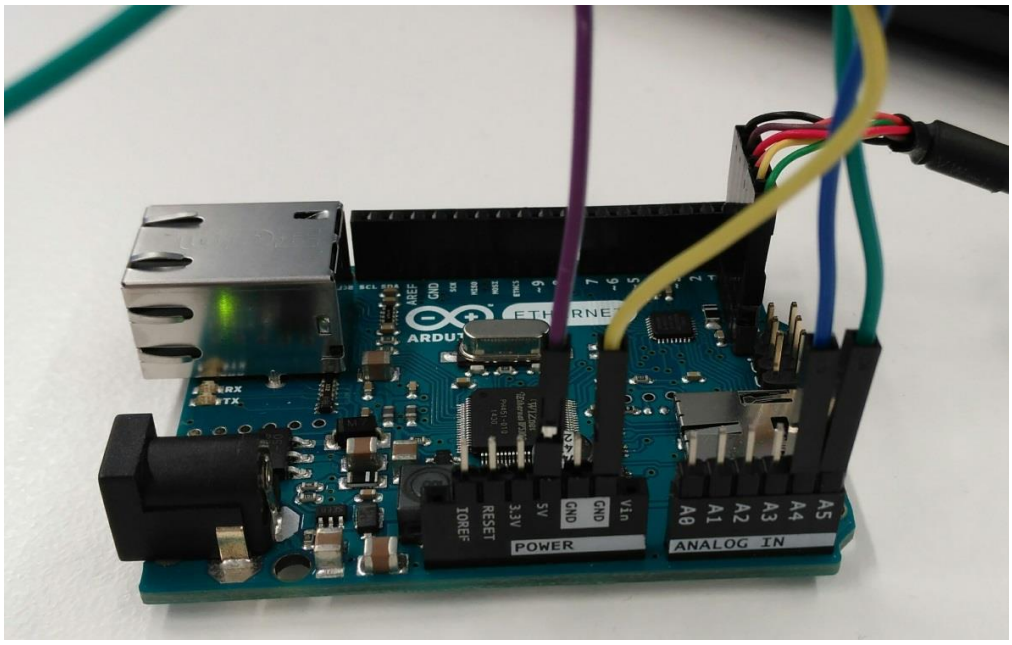
Un cop tenim el sensor muntat, passem a la connexió dels seus pins.

El sensor el connectarem, seguint les instruccions:

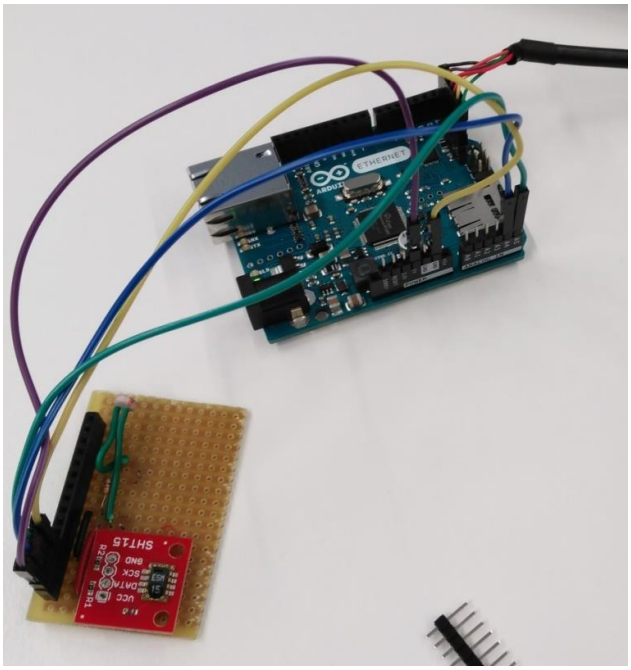
- VCC: alimentació, al pin 5V del microcontrolador [cable lila]
- DATA: al pin A5 del microcontrolador [cable blau]
- SCK: sync del clock, al pin A4 del microcontrolador [cable verd]
- GND: terra, al pin GND del microcontrolador [cable groc]



Connexió al microcontrolador segons comentat:



Un cop muntat i connectat al microcontrolador:



Si ens fixem en la imatge, el cable lila de VCC va al pin de 5V del microcontrolador; el cable blau de DATA, al pin analògic A4; el cable verd de SCK (sync), al pin A5; i el groc de GND (ground), al GND.

7.2.2.2 Sensor de llum (LDR)

Aquest sensor és més especial. La fotoresistència LDR és un sensor de tipus resistiu.

És un tipus de resistència que canvia de valor segons la quantitat de llum que incideixi sobre ella.

Les entrades analògiques *d'Arduino* poden mesurar voltatges entre 0-5V, però no valors de resistència del LDR.

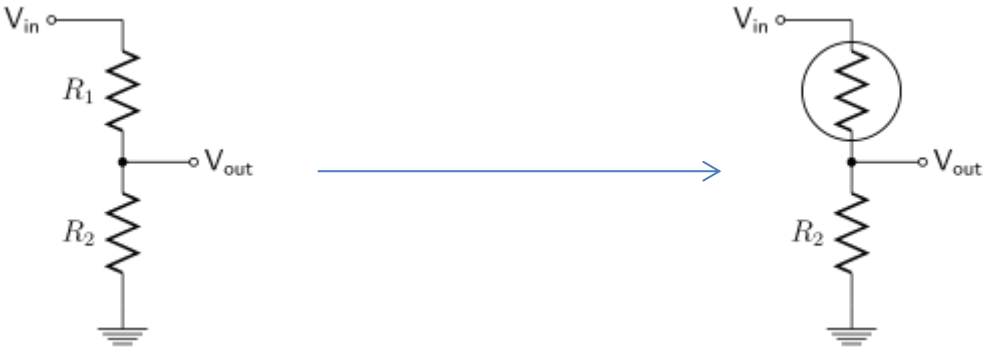
Per poder mesurar la quantitat de llum utilitzant un LDR necessitem convertir la seva resistència a un voltatge.

El circuit més senzill per fer-ho és el divisor de tensió, també conegut com a divisor de voltatge, que consisteix en dos resistències connectades en sèrie a les quals s'aplica un voltatge al extrem Vin. Al conduir corrent a través d'aquestes

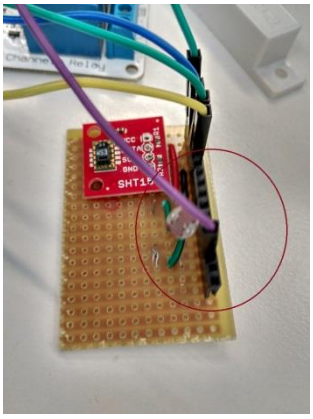
dos resistències, es produeix un voltatge en el punt on s'uneixen, Vout, valor que es pot determinar mitjançant la fórmula:

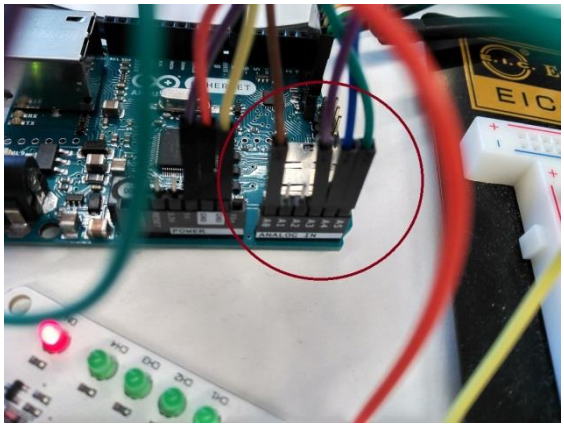
$$V_{out} = V_{in} \cdot \frac{R_2}{R1 + R2}$$

En el nostre cas amb el sensor LDR, construirem un divisor de tensió on R1 és el sensor LDR i R2 una resistència que inclourem en el circuit (21):



Ho hem posat a la mateixa plaqueta que el de temperatura. L'únic que tenim que fer un cop soldat a la placa, és connectar-hi un cable de dades i aquest al microcontrolador.



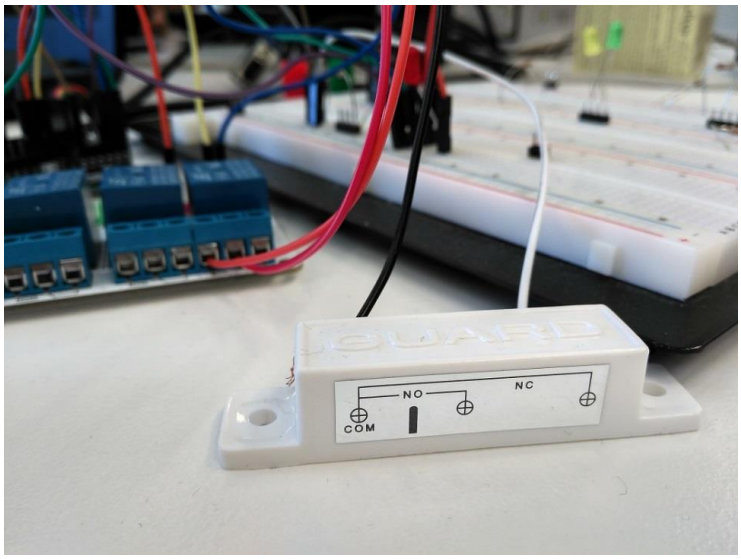


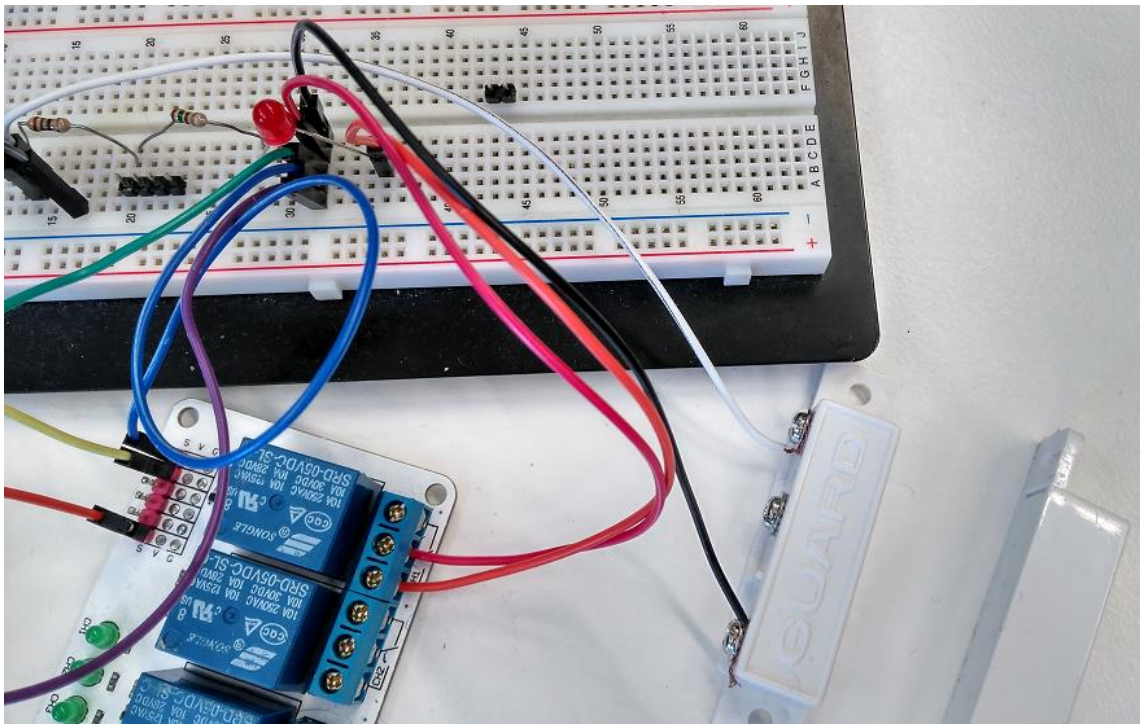
Si ens fixem amb les imatges, veurem que el cable lila connectat al sensor de llum, va al pin A3 del microcontrolador.

7.2.2.3 Sensor de porta magnètica

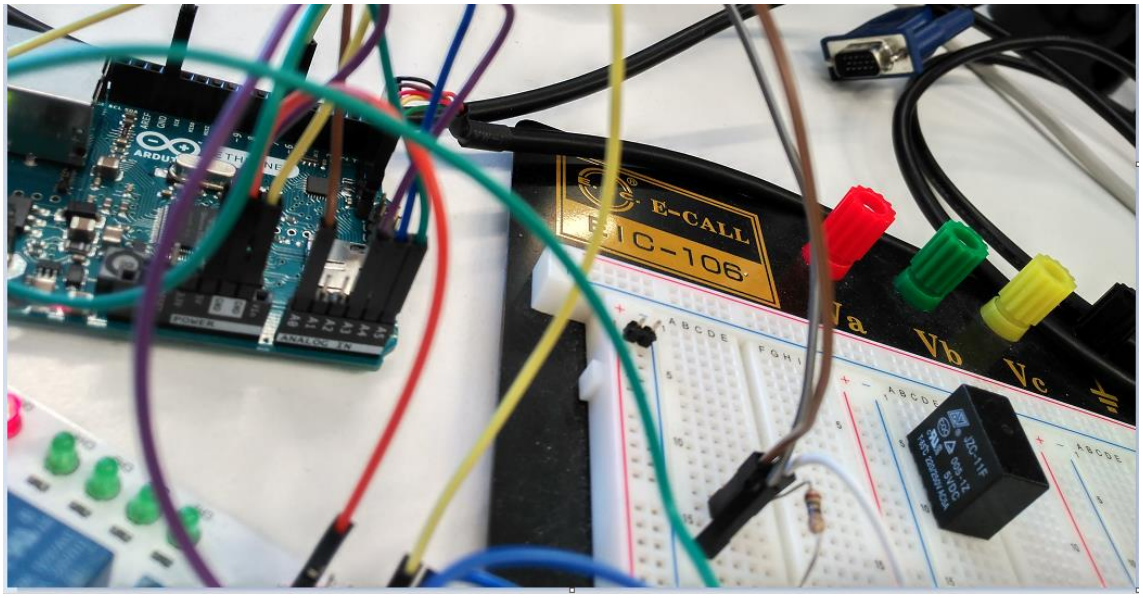
El sensor de porta magnètica, és un dels més fàcils de muntar ja que no cal soldar res.

Simplement connectem el cable, pelat, en contacte amb cada un de les entrades de la porta magnètica, i ja ho tenim.





Si ens fixem en la imatge el cable blanc va a una fila de la *Protoboard*, on posarem un altre cable en la mateixa fila (ja que no ens arriba a la placa directament) per connectar-lo al pin analògic A0 del microcontrolador, i així poder veure l'estat de la porta.



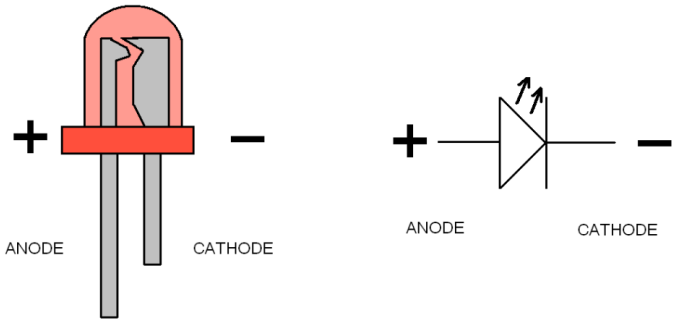
El cable negre, en canvi, el posarem a una fila de la *Protoboard* per donar-li tensió, ja que no tenim més pins de voltatge al microcontrolador.

7.2.3 Muntatge del *led* amb el relé

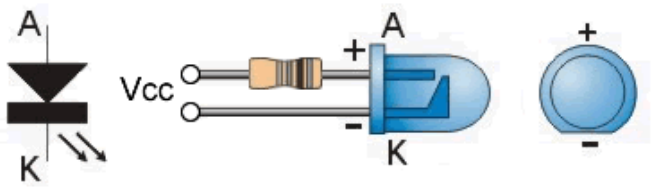
Primer de tot veurem com es munta un *led*. Un díode *led*, és un component que té polaritat, fet pel qual l'hem de connectar correctament.

Tot *led* té que dur associada en sèrie una resistència per limitar la corrent que passa per ell. Sense ella, podria cremar-se.

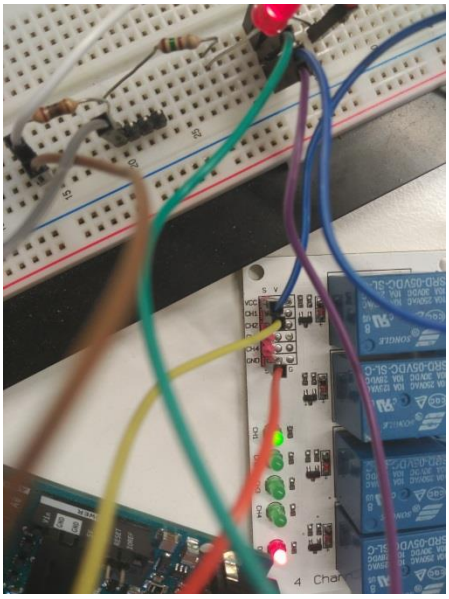
Polaritat:

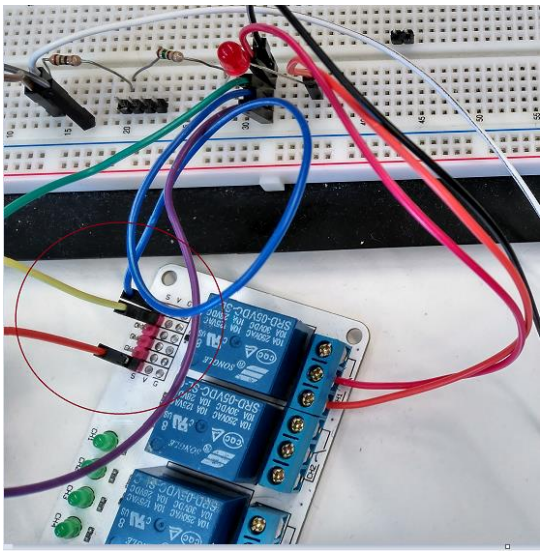


Connexió en sèrie amb la resistència (21):

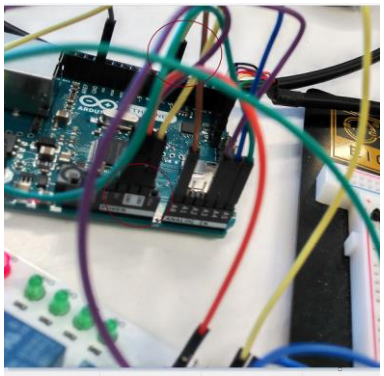


El relé també té uns pins de connexió d'entrada, que haurem de connectar al voltatge i al GND (terra)

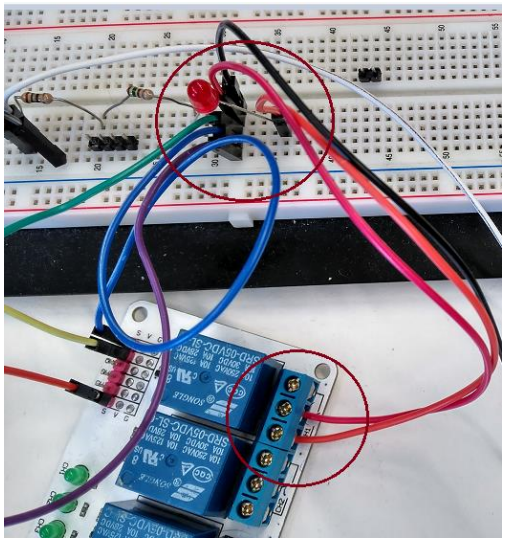




Si ens fixem en la imatge el cable vermell és el de GND, que va al segon pin de GND que té *l'Arduino*; el cable groc de dades va al pin digital D7 del microcontrolador (ja que sols enviarem 0 o 1); i el blau va a una fila de la *Protoboard* on tenim tensió, per alimentar el relé, ja que no tenim més línies de tensió, pel qual fem ús de la *Protoboard*.



En quant a la connexió del *led*, serà amb els cables de sortida. El funcionament que volem per al *led* és que si no li enviem tensió al relé, el *led* es mantingui apagat, i si li enviem, s'encengui. Llavors el que hem de fer és connectar al pin que seria el tancament del circuit (cable de color granat) a la tensió, per a que quan es tanqui el circuit, passi la tensió i s'engegui el *led*.



7.2.4 Programació del microcontrolador

Primer de tot hem d'adjuntar les llibreries necessàries per poder programar-ho.

Com ja hem dit, necessitarem enviar les dades a un servidor extern, per lo que necessitarem de connexió a internet per fer-ho. Per tant necessitem adjuntar la llibreria de "Ethernet", connectar el nostre *Arduino* a un cable *Ethernet* amb xarxa i configurar-lo amb una ip dinàmica.

També volem llençar la crida al *webservice* cada 50 segons, pel qual farem ús també de la llibreria "SimpleTimer".

Per últim, tenim el sensor SHT11, que té una llibreria especial que ja ens ho dona tot fet: "SHT1x.h"

```
#include <Ethernet.h>
#include "SimpleTimer.h"
#include "SHT1x.h"
```

Després inicialitzarem totes les variables que necessitem, donat els pins que tenim ocupats. Ho fem seguint la lògica següent:

```
//llum
int valor_llum = 0; //on capturarem el seu valor
int pin_llum = 3; //pin on es troba ubicat en el microcontrolador
```

Com el sensor de temperatura i humitat és digital, el declarem diferent per després poder llegir els seus valors.

```
//Pins pel sensor de temperatura i humitat STH15
#define dataPin  4
#define clockPin 5
SHT1x SensorHumedad(dataPin, clockPin); // Temperatura i humitat
```

En el setup del microcontrolador, li indicarem de quin tipus és el sensor (pot ser que tinguéssim més, i haguéssim d'enviar unes altres dades). A més, cridem cada 50 segons la funció sendData (envia dades de les mesures al servidor).

```
void setup() {
  tipo_sensor = 3;
  Serial.begin(9600);
  pinMode(pin_rele, OUTPUT); //configurem el relé com a sortida
  timer.setInterval(50000, sendData);
}
```

En el bucle del microcontrolador, llegim els valors de tots els sensors, els imprimim (per veure que són correctes) i els posem en un vector.

```
void loop() {
  valor_llum = analogRead(pin_llum);
  valor_puerta = analogRead(pin_puerta);
  delay(1000);
  timer.run();

  if (tipo_sensor ==3)
  {
    temperature_out = SensorHumedad.readTemperatureC();
```

```
    humidity_out = SensorHumedad.readHumidity();

    Serial.println("VALORES:");
    Serial.print(temperature_out, DEC);
    Serial.print("Cº  ||  ");
    Serial.print(humidity_out);
    Serial.print("RH%   || ");
    Serial.print(valor_llum);
    Serial.print("Lumens   || ");
    Serial.print("VALOR PUERTA: ");
    Serial.print(valor_puerta);
    if(valor_puerta > 800){
      valor_puerta2 = 0;
      Serial.println("CERRADA");
    }
    else {
      valor_puerta2 = 1;
      Serial.println("ABIERTA");
     }
     Serial.println("");
     delay(1000);
  }
  delay(1000);

   Medidas[1] = humidity_out;
   Medidas[2] = temperature_out;
   Medidas[3] = valor_llum;
  Medidas[4] = valor_puerta;
...
  Medidas[21] = 0;
}
```

Un cop tenim les dades preparades, definim la funció que enviarà les dades al servidor. Aquesta configura la interfície de xarxa del microcontrolador, prepara l'enllaç per enviar totes les dades per GET, crida al *webservice* i, depenent del què ens retorna, encén, apaga o manté en el mateix estat el *led* que simula l'aire condicionat mitjançant el donar un LOW (0) o HIGH (1) al relé:

```
void sendData () {
  Ethernet.begin(mac);
  Serial.println("connecting...");
  Serial.print("IP = ");
  Serial.println(Ethernet.localIP());

  if (client.connect(server, 80))
  {
   Serial.println("Connected!");
client.print("GET http://sensors.lemontelecom.com/sendMeasures.php?");
   //Posem en l'enllaç tots els camps dels valors dels sensors s1→s21
  for (int i =1; i< 22; i++){
    client.print("&s");
    client.print(i);
    client.print("=");
    client.print(Medidas[i],HEX);
  }
  //Posem en l'enllaç la MAC del dispositiu
  client.print("&mac=");
  for (int i =0; i< 6; i++){
    client.print(mac[i],HEX);
  }
  client.println();
  delay(2000);
  if (client.available())
  {
    Serial.println("ENTREM");
    for (int i =0; i<1 ; i++){
```

```
//en c llegim el resultat que ens retorna el webservice sendMeasures,
que ens indicarà el valor que té que tenir el relé
        char c = client.read();
        Serial.print(c);

        if(c == '1'){
            Serial.println("OBRIM LED/AIRE CONDICIONAT");
            digitalWrite(pin_rele, HIGH);
        } else if(c == '0') {
            Serial.println("TANQUEM LED/AIRE CONDICIONAT");
            digitalWrite(pin_rele, LOW);
      }  }  }
    else   Serial.println("connection failed");
    Serial.println("disconnecting.");
    client.stop();  }}
```

Un cop tenim el codi, per testejar-lo el validem i el pugem al microcontrolador. Ara ja el podem deixar funcionant i enviant la informació al *webservice*.

7.2.5 Webservices

Tenim varis *webservices* que fer:

- "sendMeasures.php": on enviarà el microcontrolador totes les mesures que reculli, les tracta, mira les alarmes i executa les accions.
- "sendEmail.php": acció enviar correu.
- "mesures_504.php": per veure les mesures de la ubicació 504.
- "mesures_pc.php": per veure les mesures de la ubicació pc.
- "grafics-iso6h.php": per veure en un gràfic les mesures d'un sensor determinat durant les últimes sis hores.
- "grafics-iso.php": per veure en un gràfic les mesures d'un sensor determinat entre dos rangs de dates determinats.

7.2.5.1 sendMeasures.php

Incloem les classes necessàries a utilitzar com la connexió amb la base de dades i la llibreria que ens ajudarà a fer la connexió amb el servidor *Asterisk* per implementar l'acció "trucar".

```
include('bd.php');

include('floAPI.php');
```

Inicialitzem variables globals, rebem per GET tots els valors d'entrada mac, i valors dels possibles sensors de s1 a s21.

```
$bd= new BD(); //connexió amb bbdd
$fetxa = date("Y-m-d H:i:s"); //data en la qual es guardarà la mesura i possible alarma
$mac = filter_var($_GET['mac'],FILTER_SANITIZE_STRING); //valor mac
```

Recorrem tots els valors que ens han entrat, mirem a quin sensor pertanyen, si aquell sensor té alarmes, i es genera l'alarma en cas de que els seus valors no compleixin els rangs correctes i no hi hagués ja una alarma d'aquell nivell per a aquell sensor. Si aquella alarma a més té una acció a fer, s'executa.

Agafem connexions que té el microcontrolador que ens ha entrat.

```
$sel_dades = "select conexion from sensores where id_microcontrolador='".$id."' ";
$res_dades = $bd->consultar($sel_dades);
```

Recorrem tots els s1→s21 per veure si corresponen a algun sensor.

```
$conexion = $res_dades[$i]['conexion'];
$sel_des = "select descripcion,tipo_sensor,id from sensores where
id_microcontrolador='".$id."' and conexion='".$conexion."'";
$res_des = $bd->consultar($sel_des);
if(!empty($res_des)) {
        $descripcion = str_replace("'","''",$res_des[0]['descripcion']);
        $tiposensor = $res_des[0]['tipo_sensor'];
```

Busquem la mesura i si té fórmula a aplicar.

```
        $sel_med = "select tipo_medida,formula_calculo from tipos_sensor where
        id='".$tiposensor."'";
        $res_med = $bd->consultar($sel_med);
        $medida = $res_med[0]['tipo_medida'];
        $x=$valor;
        $formula = $res_med[0]['formula_calculo'];
```

Si té fórmula, l'avaluem.

```
        if($formula!=null) eval("\$calcul = $formula;");
        else $calcul=round($valor,2);
        $sel_uni = "select unidades from  tipos_medida where id='".$medida."'";
        $res_uni = $bd->consultar($sel_uni);
        $unidades =$res_uni[0]['unidades'];
        $idsensor = $res_des[0]['id'];
```

Fem l'insert en la taula un cop la dada ha sigut tractada.

```
$sql_insert3 = "INSERT INTO mesures_elaborades (idMicro, idSensor ,description, value, units,
date) VALUES ('".$id."', '".$idsensor."', '".$descripcion."', '".$calcul."', '".$unidades."',
'".$fetxa."')";
```

```
$bd->modificar($sql_insert3);
```

Comprovació de si aquell sensor té alguna alarma.

```
$sel_al = "SELECT * FROM alarmas WHERE id_sensor='".$idsensor."'";
$ds_al = $bd->consultar($sel_al);
if(!empty($ds_al)) //hem de fer alguna comprobacio
{
        $min_1=$ds_al[0]['valor_min_1'];
        $min_2=$ds_al[0]['valor_min_2'];
        $max_1=$ds_al[0]['valor_max_1'];
        $max_2=$ds_al[0]['valor_max_2'];
        ...
```

Mirem si aquella alarma té alguna acció a realitzar.

```
        $sel_accio = "SELECT * FROM acciones WHERE id_alarma='".$idal."'";
        $ds_idacc = $bd->consultar($sel_accio);
```

Comprovem si el valor calculat està dins el valor de l'alarma

```
        if($calcul<=$min_1 || $calcul>=$max_1) {
                if(!empty($ds_idacc)) {
                        $Na=count($ds_idacc);
                        $grav = 0;
```

Recorrem totes les accions associades a l'alarma.

```
        for($nalarm=0;$nalarm<$Na;$nalarm++) {
                $param=$ds_idacc[$nalarm]['parametro'];
                $tipo_accion = $ds_idacc[$nalarm]['id_accion'];
                if($tipo_accion==1) {
```

Estem en l'acció "Enviar correu"

```
        if($calcul<=$min_2 || $calcul>=$max_2) {
                $grav = 2;
                $sel_al_act = "select * from log_alarmes where idalarma='".$idal."' and
                Validada=-1 and gravedad='".$grav."'";
                $al_enviada = $bd->consultar($sel_al_act);
```

Si no hi ha cap alarma.

```
                if(empty($al_enviada))
```

```
        enviar_mail("Alarma seria!", $param, $id, $idsensor, $ubi, $descripcion,
        round($calcul,2), $unidades);
else {}
```

Si el valor del sensor no és >=max2 o <=min2.

```
        $grav = 1;
        $sel_al_act = "select * from log_alarmes where idalarma='".$idal."' and Validada=-1
        and gravedad='".$grav."'";
        $al_enviada = $bd->consultar($sel_al_act);
        if(empty($al_enviada))
                enviar_mail("Alarma
                previa",$param,$id,$idsensor,$ubi,$descripcion,round($calcul,2),$unidades);
```

Un cop hem recorregut totes les accions, si no existia una alarma, la creem.

```
if(empty($al_enviada)) {
        $sql_insert_alarm="insert into log_alarmes(hora,idalarma,valor,gravedad,Validada)
        values('".date("Y-m-d H:i:s")."','".$idal."','".$calcul."','".$grav."',-1)";
        $bd->modificar($sql_insert_alarm);} ...
```

7.2.5.2 sendEmail.php

Rebem les dades per POST del destinatari, assumpte, sensor, ubicació del sensor, microcontrolador al que pertany el sensor, mesura i unitats, crea el cos del missatge en HTML, configura tots els paràmetres de la classe *Phpmailer* i crida l'enviar.

```
require_once('class.phpmailer.php');
include ("class.smtp.php");
```

recollida de paràmetres, si ens entren tots procedim

```
$email = $_POST['email'];
...
```

creem el body, el cos el missatge i cridem a la funció enviar_email

```
$cuerpo = body($sensor,$ubi,$micro,$medida,$unidades);
if(!empty($cuerpo)){
```

```
$enviado = enviar_email($email,$subject,$cuerpo);
if($enviado == 0){
    $arr = array('result' => 0,"error_email" => $enviado,"message" =>
$message,"email" => $email); //correo no enviado
    print_r(json_encode($arr)); //correo no enviado
...
```

Funció enviar email, configura tots els camps necessaris amb les dades del SMTP, usuari, contrasenya, port, remitent, direcció de retorn, destinatari, assumpte, cos del missatge, si és en HTML o no, i l'envia.

```
function enviar_email($email, $subject, $message){
    $mail = new PHPMailer();
    $mail->IsSMTP();
    $mail->SMTPAuth   = true;
    $mail->Host  = "mail.esadecreapolis.com";
    $mail->Port  = 25;
    $mail->Username   = "isoquimen.alarmes@esadecreapolis.com";
    $mail->Password   = "Ap123456";
    $mail->SetFrom('isoquimen.alarmes@esadecreapolis.com','Alarmes Sensors');
    $mail->AddReplyTo("isoquimen.alarmes@esadecreapolis.com","Sensors");
    $mail->Subject    = $subject;
    $mail->MsgHTML($message);
    $mail->addAddress($email);
    $mail->isHTML(true);

    if( !$mail->Send() )   return $mail->ErrorInfo;
    else   return 1;
```

7.2.5.3 mesures_504.php i mesures_pc.php

Creem les dades que han de sortir de cada sensor: mesura, unitats, si ha d'anar en verd groc o vermell, l'enllaç al gràfic dels valors de les seves sis últimes hores

i generem un *html* que ho inclogui tot. Posem un tros de codi pertanyent al sensor de temperatura extern.

```
$sel_s9 = "SELECT m.value FROM mesures_elaborades AS m WHERE date = (SELECT MAX(date) FROM mesures_elaborades WHERE idSensor=$sTempExt) AND idSensor = $sTempExt";
$sel_mes_s9 = "SELECT tm.unidades FROM tipos_medida as tm, tipos_sensor as ts, sensores as sens WHERE tm.id=ts.tipo_medida AND sens.tipo_sensor=ts.id AND sens.id=$sTempExt";
$val_s9 = $bd->consultar($sel_s9);
$mes_s9 = $bd->consultar($sel_mes_s9);
```

Posem a aux9 el valor del sensor, a s9_al el codi del color que toqui si hi ha alarmes, i a s9_link el seu enllaç a la gràfica de les últimes sis hores per quan hi cliquem.

```
if(isset($val_s9)){
    $aux9 = round($val_s9[0]['value'],1);
    $s9_al = hi_ha_alarmes($bd,$aux9,$sTempExt);
    $s9_link = $server.'grafics-iso6h.php?idsensor='.$sTempExt;
    $s9 = $aux9." ".$mes_s9[0]['unidades'];
```

Funció que ens retorna el valor del color en html que posarem al muntar la pantalla final.

```
function hi_ha_alarmes($bd,$valor,$sensor) {
    $sel_alarm = "SELECT valor_min_1,valor_min_2,valor_max_1,valor_max_2 FROM alarmas WHERE id_sensor='".$sensor."'";
    $alarm = $bd->consultar($sel_alarm);
    $val_al = "background-color:green";
    if(isset($alarm)) {
        $min1 = $alarm[0]['valor_min_1'];
        $min2 = $alarm[0]['valor_min_2'];
```

```
            $max1 = $alarm[0]['valor_max_1'];
            $max2 = $alarm[0]['valor_max_2'];
            if($valor>$max2 || $valor<$min2) { //ALARMA GREU - VERMELL
                    $val_al = "background-color:red";
            }
            elseif($val>$max1 || $val<$min1) { //ALERTA LLEU - GROC
                    $val_al = "background-color:yellow";
            }
            else { //NORMAL - VERD
                    $val_al = "background-color:green";
            }
    }
    return $val_al;
}
```

Finalment el muntatge del html.

```
<html>
    <table>
    <tr><td>"EXTERNAL MEASUREMENTS"</td></tr>
    <tr><td> "Temperature" <a href="<?php echo $s9_link;?>"> <input type="text"
style=<?php echo $s9_al;?> name="Temperature" value="<?php echo $s9;?>"></a>
<br><br> </td></tr>
....
```

7.2.5.4 grafics-iso6h php i grafics-iso.php

Ens entra el sensor en les últimes sis hores, i al segon, a més a més les dos dates.

Utilitzant la llibreria *RGraph* construïm una estructura amb les dades d'aquell sensor en les dates determinades.

En el de les sis hores, calculem la data inicial i la data final. Un cop ho tenim en els dos casos demanem les dades d'aquell sensor per aquells dos rangs amb la funció

```
$res=get_valores_grafico($bd, $id_sensor, $f_ini, $f_fi);
```

Aquesta funció ens construeix el vector de valors que utilitzarà *RGraph* d'aquesta forma:

```
function get_valores_grafico($bd, $id_sensor, $f_ini, $f_fi) {

$q="SELECT value,units,date FROM mesures_elaborades WHERE idSensor=".$id_sensor." AND date>='".$f_ini."' AND date<='".$f_fi."' GROUP BY date ORDER BY date ASC";
$aux=$bd->consultar($q);
$num_valores=count($aux);
$res['nvals']=$num_valores;
if ($num_valores == 0 )
{
        echo "No hay valores para generar un grafico.";
        exit();
}
aux2 = split(" ",$f_ini);
$date2 = $aux2[1];
$aux2 = split(":",$date2);
$date2 = $aux2[0].":".$aux2[1];

$res['y']="[";
$res['x']="['".$date2."',";
$res['datos']="[";
$res['xinfo']="[";
$fnext = date('Y-m-d H:i:s', strtotime('+1 hour' , strtotime($f_ini)));
```

I per cada iteració de valor que tinguem per aquell sensor anem concatenant els valors depenent de si estem en l'últim valor o no:

```
for($i=0;$i<$num_valores;$i++)
{
        $val=$aux[$i]['value'];
        $unit=$aux[$i]['units'];
        $date=$aux[$i]['date'];
```

```
$aux2 = split(" ",$date);
$date2 = $aux2[1];
$aux2 = split(":",$date2);
$date2 = $aux2[0].":".$aux2[1];
```

I ho tanquem de la forma:

```
$res['y'].="]";
$res['datos'].="]";
$res['x'].="'".$date2."']";
$res['xinfo'].="]";
```

"xinfo" és la etiqueta que es mostrarà en el gràfic al posa'ns sobre un valor qualsevol

"x" són els valors de la barra horitzontal.

"y" són els valors de la barra vertical .

"datos" són els valors que han d'aparèixer en el gràfic.

Un cop tenim tots els resultats a $res, els dividim cada un en una variable.

```
$res=get_valores_grafico($bd, $id_sensor, $f_ini, $f_fi, $dias);
$valores=$res['y'];
$etiquetas_x=$res['x'];
$nx=$res['nvals'];
$colores= "['#00A000','#FFFF00','#FF8000','#FE0000']";
$valint = $res['xinfo'];
```

Un cop tenim totes les dades per separat, creem la taula d'aquesta forma:

```
$tabla="
<canvas id='cvs' width='2000' height='400'>[No canvas support]</canvas>";
$script="
 <script>
      window.onload = function ()
      {
```

```
var line = new RGraph.Line('cvs', ".$valores.");
line.Set('chart.labels', ".$etiquetas_x.");
line.Set('chart.tooltips', ".$valint.");
line.Set('ymin', ".$ymin.");

line.Set('ymax', ".$ymax.");
line.Draw();
```

I després utilitzem una plantilla que tenim definida i li substituïm els valors pels nostres.

```
$path="./plantillas/plantilla.php";
if (is_file($path) ) {
 $f = fopen($path, "r");
 while (($l = fgets($f, 4096)) !== false ) {
   $original="#tabla#";
   if (strpos($l, $original) > 0) { echo str_replace($original,$tabla,$l); continue; }
   $original="#link_anterior#";
   if(strpos($l,$original)> 0) { echo str_replace($original,$link_anterior,$l); continue; }
   $original="#link_posterior#";
    if(strpos($l, $original) > 0) { echo str_replace($original,$link_posterior,$l); continue;}
```

Finalment imprimim la plantilla.

```
echo $l;
```

7.2.6 Connexió amb Asterisk

Tenim un *Asterisk* instal·lat en un altre servidor. El que hem fet és, mitjançant la llibreria *floAPI.php*, connectar-nos a aquest servidor *Asterisk* i enviar-li la comanda de la trucada que volem realitzar.

Per fer-ho, primer de tot necessitem d'un usuari i contrasenya per crear la connexió amb floApi. El creem en el fitxer *manager.conf* del servidor *Asterisk*, que es troba ubicat a "/usr/local/etc/asterisk/".

asterisk1# cat manager.conf

```
[general]
displaysystemname = yes
enabled = yes
webenabled = yes
port = 5038
.....
[cess]
secret = pfc2008
deny=0.0.0.0/0.0.0.0
permit=192.168.7.65/255.255.255.255
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config
...
```

Un cop fet això, hem gravat dos missatges de veu: un per l'alarma lleu i un per l'alarma greu. Ho hem fet mitjançant una extensió que hem configurat per gravar missatges dins de extensions.conf, ubicat a "/usr/local/etc/asterisk/":

asterisk1# ee extensions.conf

```
;grabar

exten => 8499,1,Goto(grabar,205,1)
```

I definim el contexte grabar més avall:

```
[grabar]
exten => 205,1,Answer
exten => 205,2,Wait(2)
exten => 205,3,Record(GRABACIO%d:gsm)
exten => 205,4,Wait(2)
exten => 205,5,Playback(${RECORDED_FILE})
exten => 205,6,Wait(2)
exten => 205,7,Hangup
```

Això ens deixa els missatges a "/usr/local/share/asterisk/sounds/", amb un nom automàtic. Els hem ubicat per hora de la gravació i li hem canviat els noms a alarma-lleu.gms i alarma-greu.gms.

Després, creem el context que utilitzarem per trucar, dins el fitxer *extensions.conf* també:

```
asterisk1# ee extensions.conf

    [avisos]
        exten => previous,1,set(sip_CODEC=g729)
        exten => previous,n,Set(CALLERID(num)=5005)
        exten => previous,n,Answer
        exten => previous,n,wait(6)
        exten => previous,n,Read(tecla1,alarma-lleu,1,,60,4)
        exten => previous,n,Gotoif($[${tecla1}=1]?avisos,hangup,1)
        exten => previous,n,wait(1)
        exten => previous,n,Hangup
        exten => serious,1,set(sip_CODEC=g729)
        exten => serious,n,Set(CALLERID(num)=5005)
        exten => serious,n,Answer
        exten => serious,n,wait(6)
        exten => serious,n,Read(tecla1,alarma-greu,1,,60,4)
        exten => serious,n,Gotoif($[${tecla1}=1]?avisos,hangup,1)
        exten => serious,n,wait(1)
        exten => serious,n,Hangup
        exten => hangup,1,Hangup()
```

Anem a implementar l'acció de trucar. Consistirà en executar un *originate* passant-li la línia del context [avisos] on hem d'anar (depenent del nivell de gravetat de l'alarma), i el número que volem trucar, amb un 0 davant per indicar que és un .número extern i no una extensió

```
switch($level){
    case 1:  $command="originate Local/previous@avisos extension ".$param."@nivell4";
```

```
                break;
        case 2: $command="originate Local/serious@avisos extension 0".$param."@nivell4";
                break;
        default: break;
}
$astcc = new floAPI("cess","pfc2008","192.168.6.12");
$astcc->events_toggle("ON");
$content ="";
$content = $astcc->request("COMMAND",array("COMMAND"=>$command));
```

Finalment, ho incloem tot en el *webservice* "sendMeasures.php", en l'acció de trucar cridem a la funció call amb el paràmetre del número al que volem trucar, i el nivell de gravetat: avís o alarma greu. Generarem la comanda a executar: "*originate*" amb els paràmetres corresponents, i finalment amb un usuari i contrasenya correctes en el servidor *d'Asterisk*, creem un objecte *floApi* i li passem la comanda a executar.

```
function call($param,$level){
  $origen = 5005;
  switch($level){
    case 1:
        $command="originate Local/previous@avisos extension 0".$param."@nivell4";
        break;
    case 2:
        $command="originate Local/serious@avisos extension 0".$param."@nivell4";
        break;
    default: break;
  }
  $astcc = new floAPI("cess","pfc2008","192.168.6.12");
  $astcc->events_toggle("ON");
  $content ="";
  $content = $astcc->request("COMMAND",array("COMMAND"=>$command));
```

```
 $t=0;
 ....
 return $content;
}
```

7.2.7 Connexió amb plataforma SMS

Per realitzar l'acció d'enviar un missatge, cridem a la funció enviar_SMS amb els paràmetres d'assumpte, número de mòbil, microcontrolador, sensor, ubicació, contingut del missatge, mesura i unitats.

Com tenim una compta en la plataforma de Sit Consulting, enviem el contingut del missatge, amb un id de client fixat i contrasenya, a la *url* mitjançant *curl* d'on tenim l'altre *webservice* que ens farà les comprovacions necessàries i s'hi connectarà per envia'l.

```
function enviar_SMS($asun,$param,$id,$idsensor,$ubi,$descripcion,$calcul,$unidades)
{
	$usuario="e-isoquimen";
	$pass="Yws872a";
	$idCliente="103";
	$fecha=date('Y-m-d');
	$des=$param;
	$text="El ".$descripcion."  actualment mesura  ".$calcul.' '.$unidades;
	$aux=md5("$pass:$fecha:$usuario:$idCliente");

	$ch = curl_init();
	$fields = array(
		'x' => $aux,
		'c' => $idCliente,
		't' => $text,
		'd' => $des
	);
	curl_setopt($ch,CURLOPT_URL,
"http://operador.lemontelecom.com/envio_sms_integracion_sin/envio_sms_integracion_si
n.php");
	curl_setopt($ch,CURLOPT_POST, 1);
	curl_setopt($ch,CURLOPT_POSTFIELDS,$fields);
```

```
    $result = curl_exec($ch);
    curl_close($ch);
}
```

En envio_sms_integracion_sin rebem la informació del client i missatge, i fem tota la comprovació de si el client que intenta enviar el missatge existeix en la base de dades, té saldo, etc.

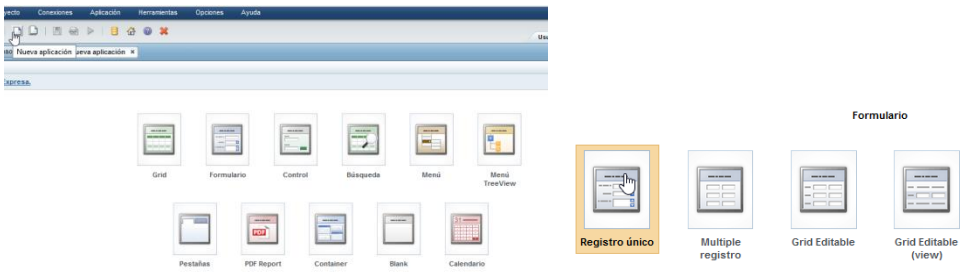
Un cop ha passat totes les comprovacions, es connecta a Sit Consulting mitjançant *curl* de nou per fer l'enviament:

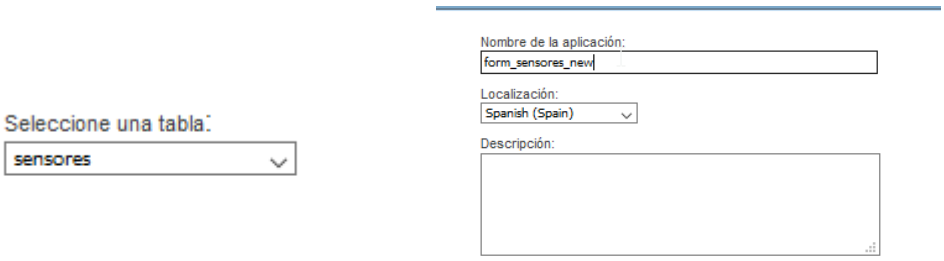
```
$curl = &new Curl_HTTP_Client();
$post_data = array('Text' => $texto, 'Destinations' => $destino, 'Login' => $login, 'Password'
=> $password, 'IdCuenta' => $account, 'Remitente' => $remitente);
$html_data= $curl-
>send_post_data("http://prmanager.sitconsulting.biz/Gateway/MT_ClienteExterno.aspx",
$post_data);
```

7.2.8 Scriptcase

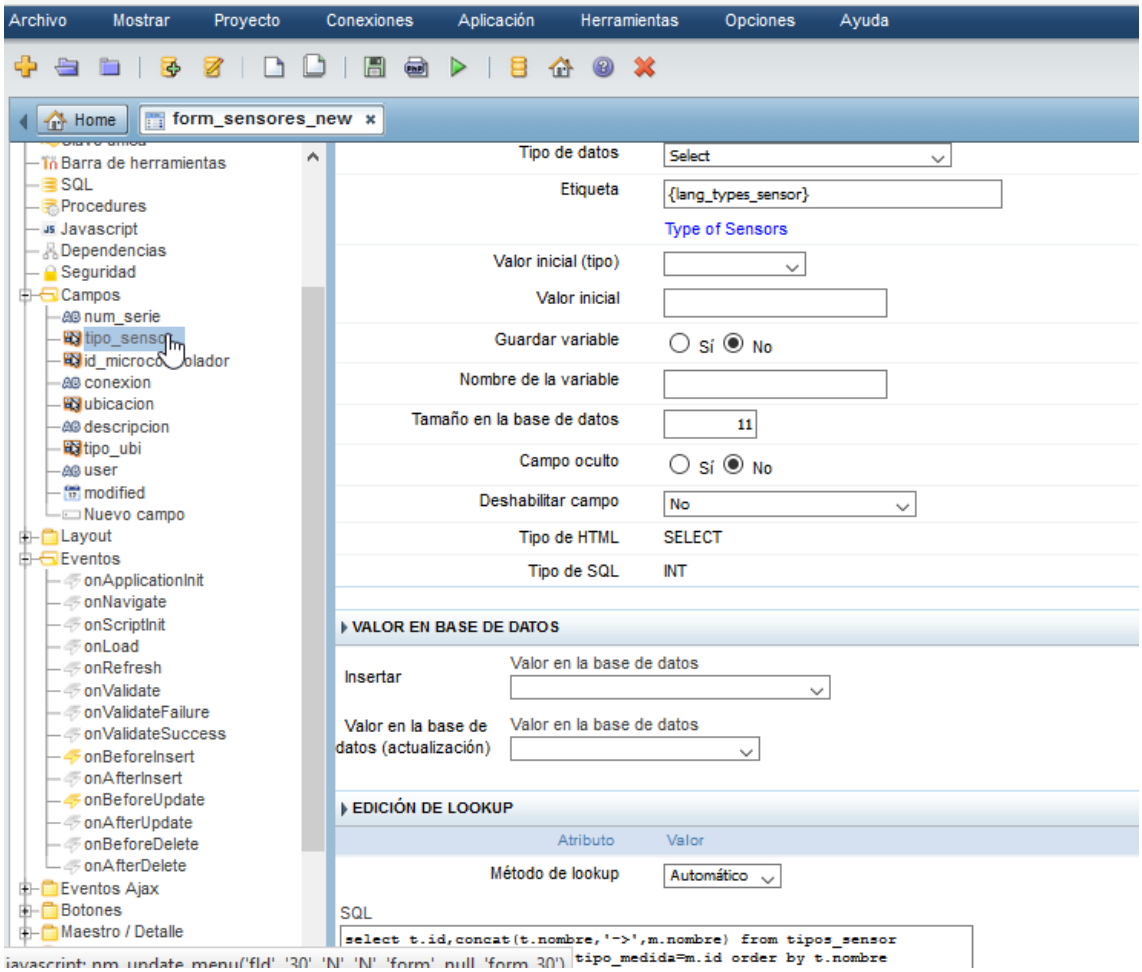
Des de *Scriptcase* creem els formularis i grids necessaris, junt amb els menús, pàgines de control, etc.

Per exemple veiem sota com hem creat un formulari per la creació d'un nou sensor. Creem una nova aplicació, escollim el tipus de formulari, seleccionem la taula a la qual anirà lligat, i un nom.

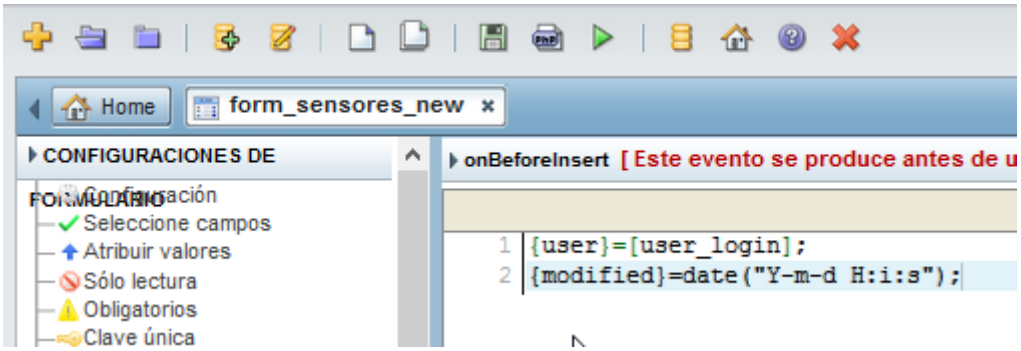




Un cop tenim l'aplicació creada, editem els camps assignant-los els valors que volem. Per exemple, quan un usuari creï un nou sensor, volem que li sigui el més fàcil possible i que hi hagi el mínim error a l'hora d'inserir els camps, així que posarem certs camps de tipus *select* i fixarem el seu valor al contingut d'altres taules.



Hi ha certs formularis als quals abans/després d'inserir el registre, actualitzar-lo o eliminar-lo, voldrem fer alguna comprovació o simplement donar valor a certs camps:



7.2.9 Triggers

Una de les peticions del client, és que hi hagi un registre de tots els canvis produïts des del *backoffice*, per poder tenir controlat si algú ha fet algun canvi i quan.

Ho hem implementat a nivell de *triggers* ja que és lo més òptim.

Tot anirà a la taula "Audit".

7.2.9.1 Trigger per l'insert d'alarmes

```
delimiter |

    CREATE TRIGGER `ins_alarmas` AFTER INSERT ON `alarmas`
    FOR EACH ROW BEGIN
    INSERT INTO `Audit` SET `Type`="I", `TableName`="alarmas",
    `PrimaryKeyField`="id", `PrimaryKeyValue`=NEW.id, `FieldName`="Alarm Id",
    `NewValue`=NEW.id, `UpdateDate`=NEW.modified,
    `UserNamevarchar`=NEW.user;
    END;
    |

delimiter ;
```

7.2.9.2 Trigger per el delete d'alarmes

```
    CREATE TRIGGER del_alarmas BEFORE DELETE ON alarmas
    FOR EACH ROW INSERT INTO
    Audit(Type,TableName,PrimaryKeyField,PrimaryKeyValue,UpdateDate,UserN
    amevarchar) VALUES('D', 'alarmas','Id',OLD.id,OLD.modified,OLD.user);
```

7.2.9.3 Trigger per l'update d'alarmes

```
    delimiter |

    CREATE TRIGGER `upd_alarmas` BEFORE UPDATE ON `alarmas`
    FOR EACH ROW BEGIN
    IF (NEW. nombre <> OLD.nombre) THEN
    INSERT INTO `Audit` SET `Type`="U", `TableName`="alarmas",
    `PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Name",
    `OldValue`=OLD.nombre, `NewValue`=NEW.nombre,
    `UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
    END IF;
    IF (NEW.id_sensor <> OLD.id_sensor) THEN
     INSERT INTO `Audit` SET `Type`="U", `TableName`="alarmas",
    `PrimaryKeyField`="id",  `PrimaryKeyValue`=OLD.id,
    `FieldName`="Sensor", `OldValue`=OLD.id_sensor,
    `NewValue`=NEW.id_sensor, `UpdateDate`=NEW.modified,
    `UserNamevarchar`=NEW.user;
END IF;
IF (NEW.tipo_alarma <> OLD.tipo_alarma) THEN
 INSERT INTO `Audit` SET `Type`="U", `TableName`="alarmas",
`PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Alarm Type",
`OldValue`=OLD.tipo_alarma, `NewValue`=NEW.tipo_alarma,
`UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
END IF;
IF (NEW.valor_min_1 <> OLD.valor_min_1) THEN
 INSERT INTO `Audit` SET `Type`="U", `TableName`="acciones",
`PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Minimum
value 1", `OldValue`=OLD.valor_min_1, `NewValue`=NEW.valor_min_1,
`UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
END IF;
IF (NEW.valor_min_2 <> OLD.valor_min_2) THEN
 INSERT INTO `Audit` SET `Type`="U", `TableName`="acciones",
`PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Minimum
value 2", `OldValue`=OLD.valor_min_2, `NewValue`=NEW.valor_min_2,
`UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
END IF;
IF (NEW.valor_max_1 <> OLD.valor_max_1) THEN
 INSERT INTO `Audit` SET `Type`="U", `TableName`="acciones",
`PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Maximum
value 1", `OldValue`=OLD.valor_max_1, `NewValue`=NEW.valor_max_1,
`UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
END IF;
IF (NEW.valor_max_2 <> OLD.valor_max_2) THEN
 INSERT INTO `Audit` SET `Type`="U", `TableName`="acciones",
`PrimaryKeyField`="id", `PrimaryKeyValue`=OLD.id, `FieldName`="Maximum
value 2", `OldValue`=OLD.valor_max_2, `NewValue`=NEW.valor_max_2,
`UpdateDate`=NEW.modified, `UserNamevarchar`=NEW.user;
END IF;
END;
|
```

delimiter ;

(Adjuntem en els annexos la resta)

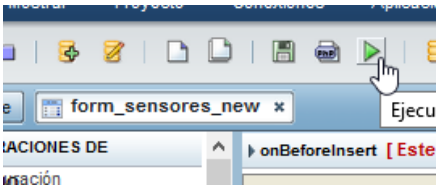
8 Proves

A continuació es mostra el joc de proves que s'ha realitzat de l'aplicació per a comprovar la correctesa d'aquesta. Totes les proves es fan partint del menú principal.

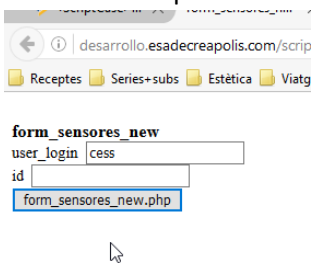
Nota: Només hi apareixen les mes rellevants, sense incloure les que es tracten del mateix control o funcionalitat.

8.1 Backoffice

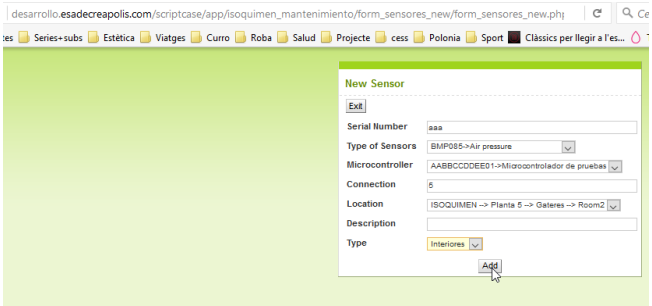
Quan creem un formulari, el testegem primer des de desenvolupament mitjançant el botó d'executar que veiem en la imatge següent:



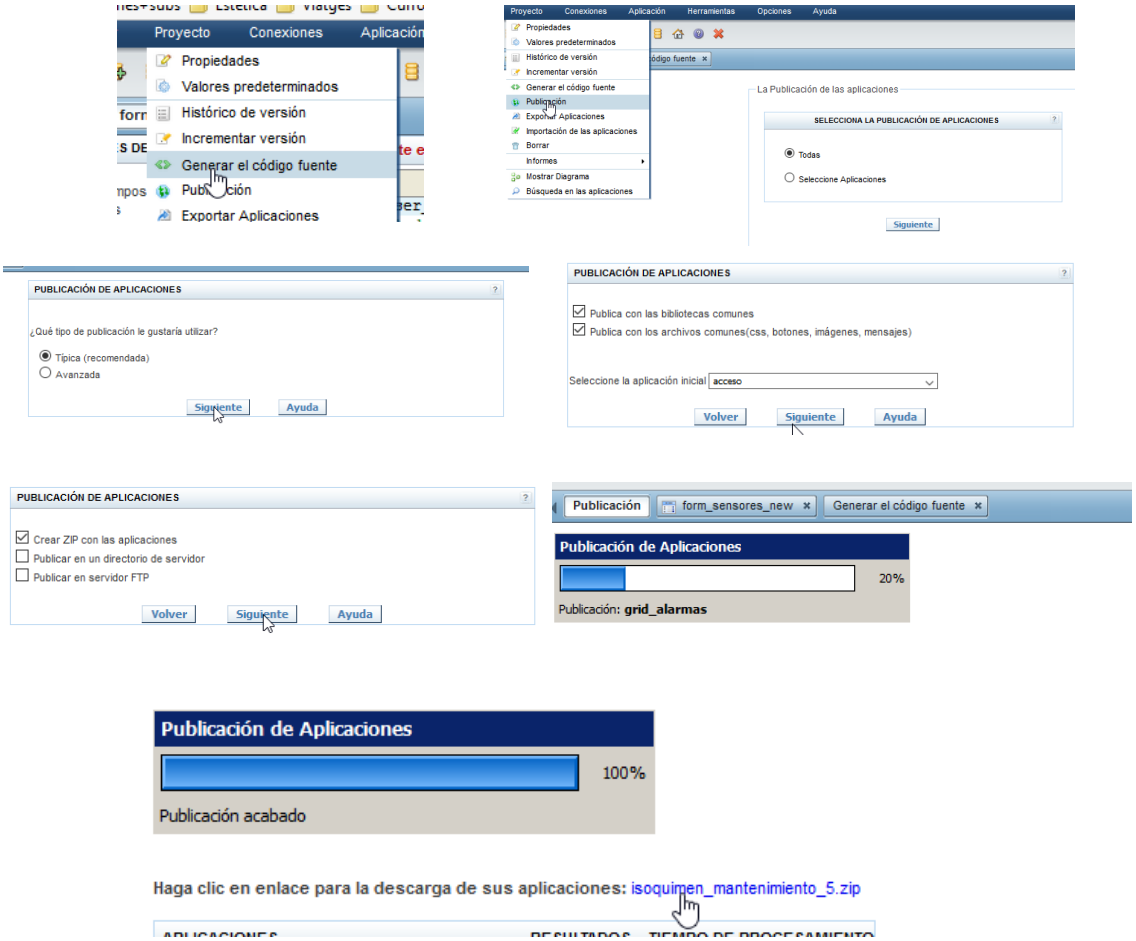
Ens obrirà una nova pestanya amb la prova. Si aquell formulari necessita paràmetres entrants, els haurem d'omplir:



Finalment ens obrirà el formulari, que omplirem. Quan li donem al botó "Afegir", veurem si tot va bé o ens dóna algun error. Pot ser que a l'hora de donar-li al botó d'executar, si hi ha errors en el codi, ens salti un avís.



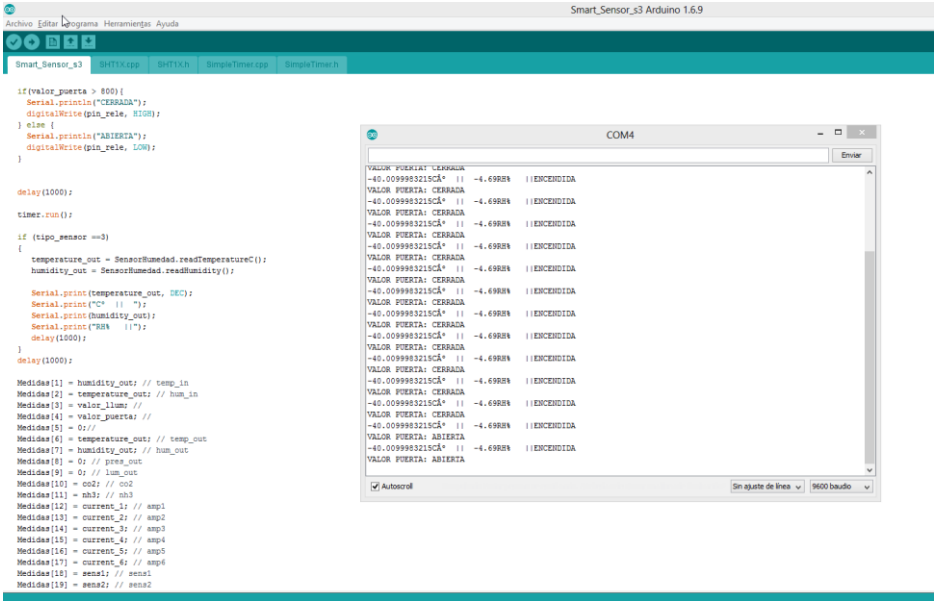
Un cop els tenim tots testejats es puja a un entorn de producció. Per fer això generem el codi font de tot el codi, i un cop el tenim tot compilat fem un desplegament del projecte.



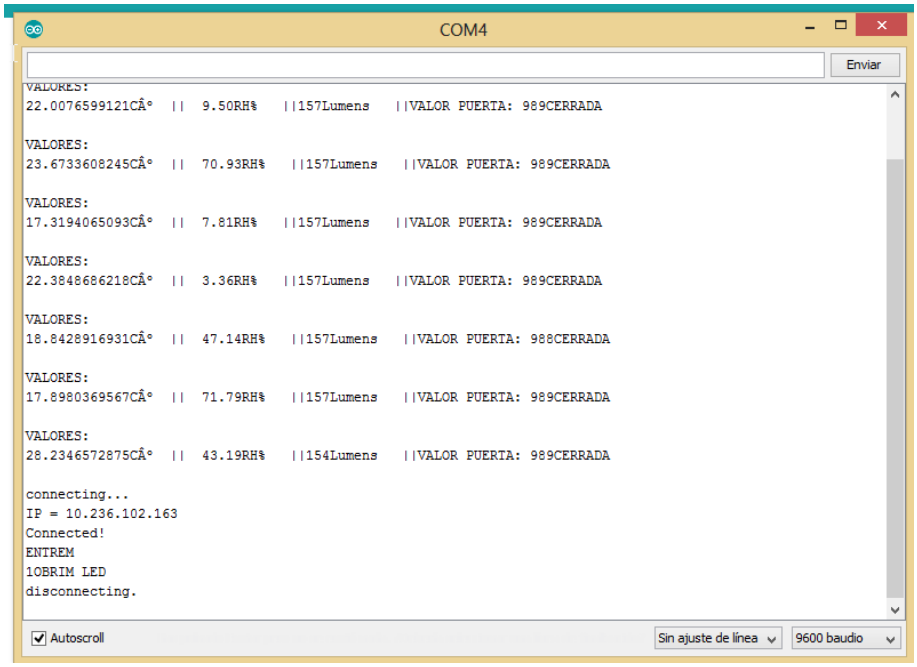
Ens baixem el fitxer generat amb tot el codi estructurat, el pugem per FTP on tinguem el servidor de producció i el descomprimim al directori que tinguem el projecte.

8.2 Sensors

A mesura que hem anat muntant els sensors al microcontrolador, hem anat fent proves amb el codi de *l'Arduino* i la seva consola, per veure que els capturava bé.



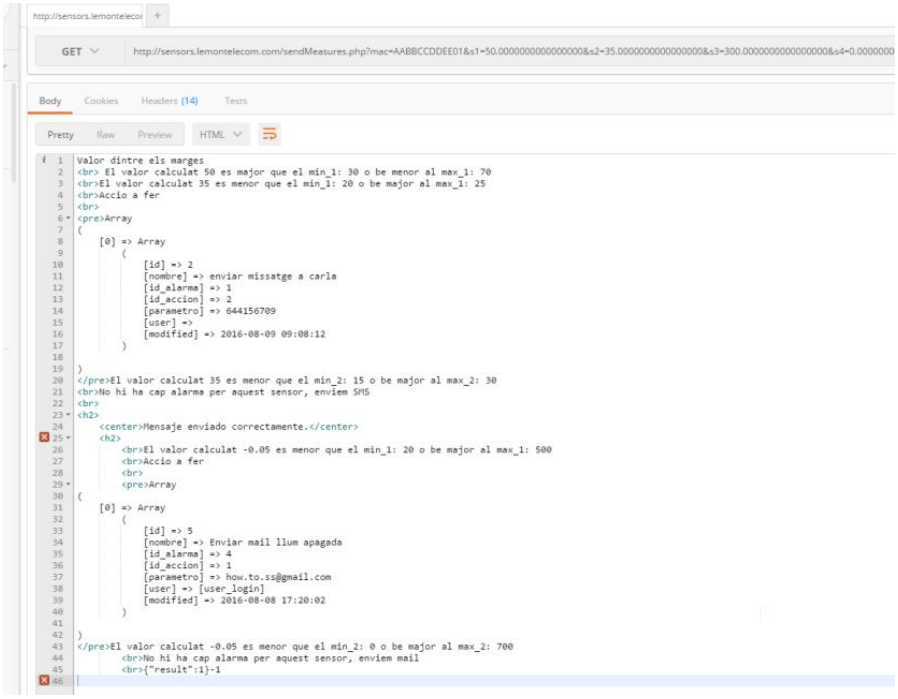
Aquesta imatge que veiem és de les proves dels sensors de temperatura, humitat, llum i porta magnètica. Hem imprès els seus valors.



8.3 Enviament de dades

S'han fet proves abans de connectar el microcontrolador al *webservice* de que aquest funcioni. Per fer les proves del *webservice* de que rebi les dades i les tracti (sendMeasures.php), hem utilitzat l'eina *Postman*.

Aquesta eina simula una crida a un *webservice* que tinguem online, i li podem passar els paràmetres de diferents maneres.

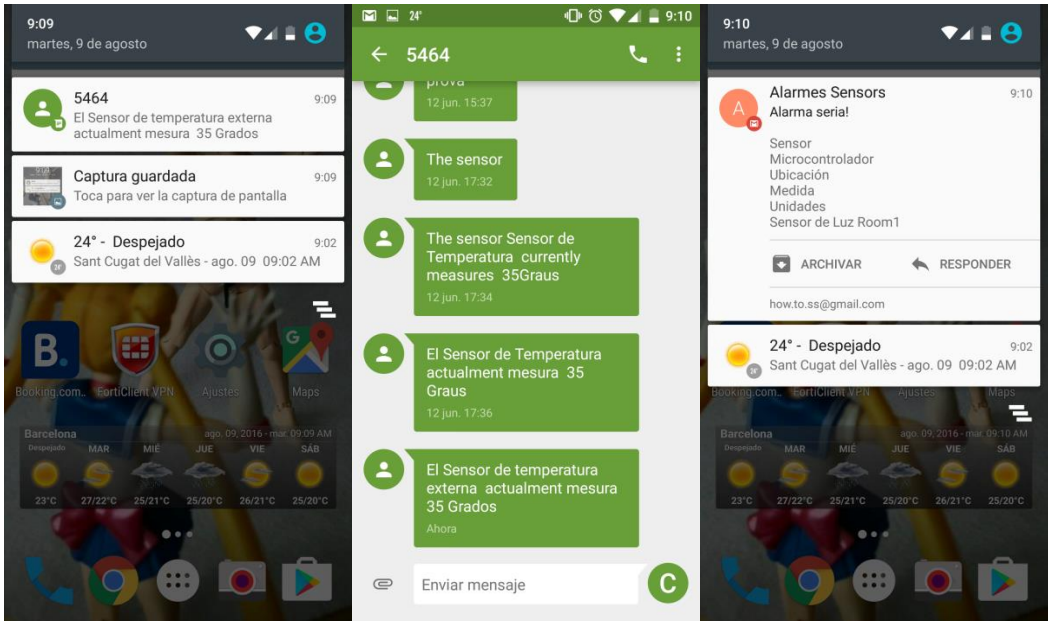


En aquesta crida veiem que hem fet una prova del *webservice* "*sendMeasures.php*", amb els paràmetres:

- Sensor d'humitat: 50 (%)
- Sensor de temperatura: 35 (graus)
- Sensor de llum: 300 (lúmens)
- Sensor de porta magnètica: 0

Com veiem, tenim una alarma pel sensor de temperatura, i una acció associada a ella que és enviar missatge.

També tenim una altra al sensor de llum, i l'acció d'enviar correu. Veiem les captures realitzades des del dispositiu mòbil:



I les del correu amb més detall:

En el *backoffice* podem observar que ens han arribat les alarmes pertinents, i que estan pendents de validar.

9 Instal·lació i execució del sistema

9.1 Requisits

Sistema operatiu: Al ser un navegador, qualsevol és vàlid.

Connexió a internet.

9.2 Instal·lació

No és necessària cap instal·lació, ja que accedim a través d'un navegador. Sols cal tenir un navegador instal·lat: Chrome, Firefox, Safari, IE...

9.3 Execució

Accedir al enllaç: http://sensors.lemontelecom.com

10 Manual d'usuari

CONTROL DE L'ENTORN EN ANIMALS

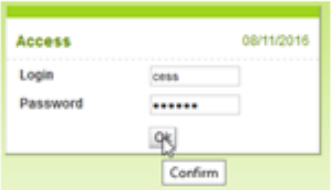
[MANUAL D'USUARI]

ÍNDEX

1. INTRODUCCIÓ

Benvinguˇ/da a la aplicació online per controlar les mesures d'un entorn animal.
Si ja ets usuari, per accedir-hi entra a:http://sensors.lemontelecom.com

2. INICIALITZAR SESSIÓ

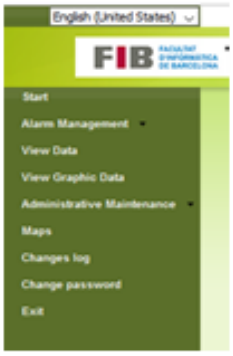


Primer de tot hauràs d'inicialitzar sessió amb l'usuari i contrasenya que t'haurà facilitat un administrador.

*Caldrà que tinguis connexió a Internet.

3. MENÚ PRINCIPAL

Accediràs a la pantalla on pots



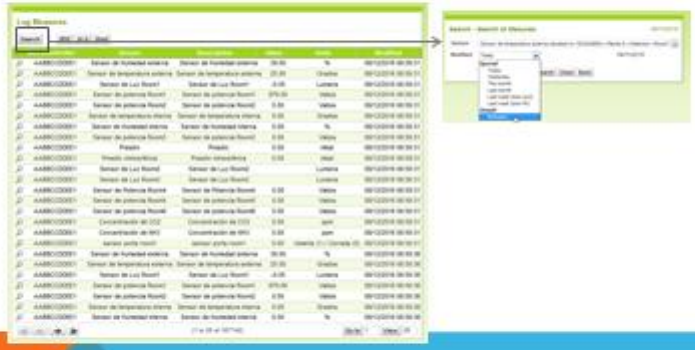
1. Veure les mesures dels sensors d'una ubicació
2. Veure i administrar les alarmes
3. Veure el registre històric dels sensors
4. Veure en un gràfic els valors d'un sensor en unes dates determinades
5. Gestionar els microcontroladors, sensors, mesures, etc.
6. Canviar el password
7. Desconnectar la sessió

4 VEURE DADES SENSORS PER UBICACIÓ

Per veure les dades dels sensors d'una ubicació, primer escollim la ubicació i al donar-li al ok ens mostra els valors de tots els sensors. Si cliquem sobre un valor podrem veure les dades de les seves últimes 6 hores en un gràfic.

5. VEURE REGISTRE HISTÒRIC SENSORS

Anant a "Veure dades" ens mostra totes les dades dels sensors de més actuals a més antigues. Si ens interessa buscar un sensor o data concreta, podem filtrar per sensor i data.



6. VEURE DADES SENSOR EN GRÀFIC EN UN PERÍODE DE TEMPS

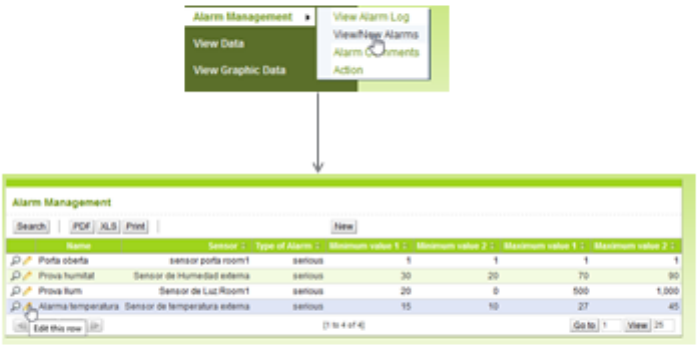
L'opció "Veure dades en un gràfic", pots escollir un sensor i un període de dates en les quals vols veure els seus valors.

7. CONTROL ALARMES, COMENTARIS, ACCIÓNS PER ALARMES...

Per veure el log d'alarmes, anem a l'opció del menú "Manteniment d'alarmes", "Veure registre", et demana si vols veure les alarmes validades o les no validades, i un cop tens el llistat, les pots editar validant-les, posant-los-hi comentaris, etc.

7. CONTROL ALARMES, COMENTARIS, ACCIÓNS PER ALARMES...

Per crear, editar o eliminar alarmes ho pots fer des del menú següent, modificant això els valors mínims i màxims d'alarma.



8. MANTENIMENT MICROCONTROLADOR, SENSORS, MESURES...

Des del menú "Manteniment Administratiu", pots gestionar els microcontroladors, sensors, mesures, tipus de sensors, etc. En la imatge veiem com veure els sensors, editar-los i crear un de nou.

9. CANVIAR PASSWORD

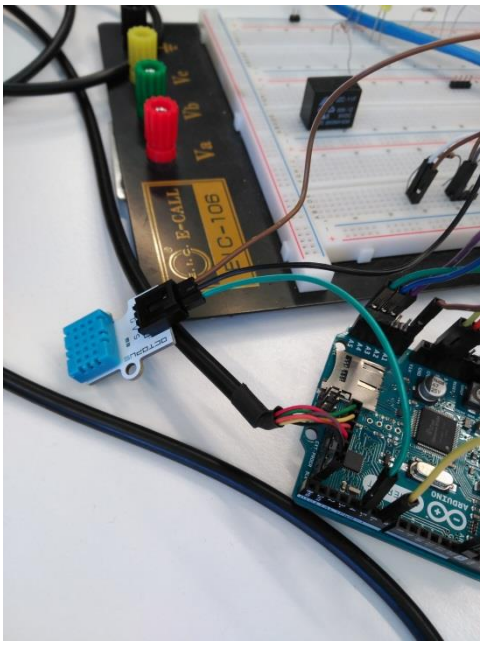
Pots canviar el password anant al menú "Canviar password", i posant el teu password actual, el nou password i la seva confirmació.

10. DESCONNECTAR

Per sortir de la sessió actual de l'usuari has d'anar a "Sortir", i et redirigirà a la pantalla d'accés inicial.

11 Problemes trobats

Enmig de les proves de dades dels sensors, el sensor de temperatura i humitat SHT15, va començar a fallar. Donava valors negatius fixes. Un cop comprovat que tots els cables estesin correctament connectats i haver reiniciat el microcontrolador varis cops, he decidit canviar de sensor per un d'un altre model més econòmic, el DHT11.



L'hem connectat a *l'Arduino* al pin digital D5, i el ground i el voltatge a unes fila que tenim a la *Protoboard* a falta de més pins a la placa. Ens baixem la llibreria corresponent "SimpleDHT.h" i "SimpleDHT.cpp", la incloem en el codi, i fem la crida seguint les instruccions d'aquesta llibreria:

```
int pinDHT11 = 5;
SimpleDHT11 dht11;
byte temperature = 0;
byte humidity = 0;
```

En la funció de loop ens imprimim els valors per veure'ls en les proves:

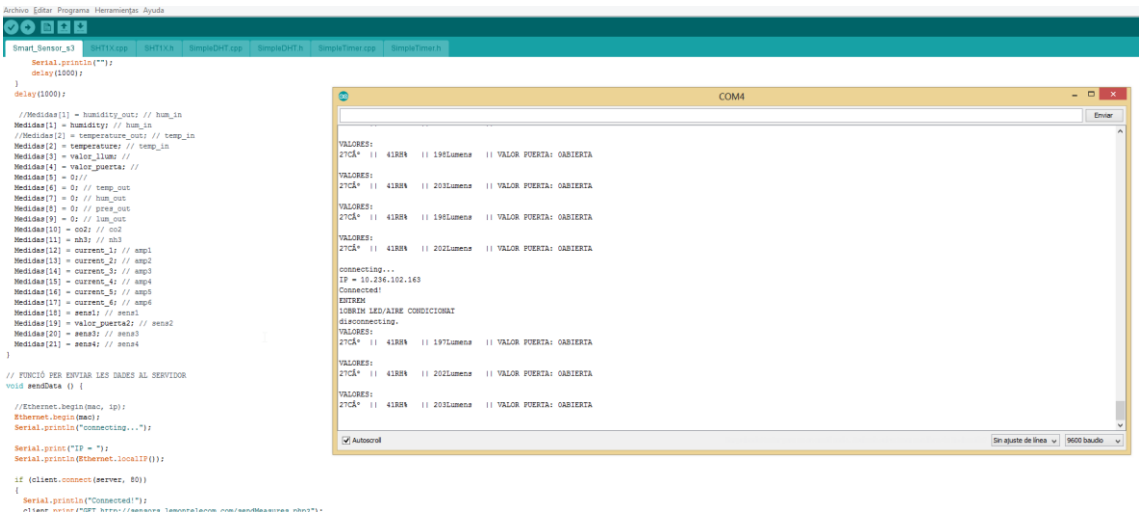
```
Serial.println("VALORES:");
    if (dht11.read(pinDHT11, &temperature, &humidity, NULL)) {
        Serial.print("Read DHT11 failed.");
```

```
} else{
  Serial.print(temperature);
  Serial.print("Cº  ||  ");
  Serial.print(humidity);
  Serial.print("RH%   || ");
}
```

I en la funció sendData() passem els nous valors:

```
Medidas[1] = humidity;

Medidas[2] = temperature;
```



12 Planificació real i valoració econòmica

Com era d'esperar hi hagut desviacions en la planificació del projecte.

Fase inicial (170h)		
	Definició del producte	25
	Decisió de tecnologies i compra material necessari	40
	Formació	90
	Redacció memòria	15
Fase planificació (85h)		
	Planificació i anàlisis de requisits	25
	Especificació	30
	Redacció memòria	20
Fase execució (305h)		
	Muntatge dels sensors	25
	Programació microcontrolador	30
	Disseny del backoffice i webservices	40
	Implementació backoffice	40
	Implementació webservices i triggers	105
	Proves	40
	Redacció memòria	25
Fase tancament (70h)		
	Retocs	30
	Manual d'usuari	10
	Finalitzar memòria	30
		Total: 630h

S'ha necessitat més temps a la fase de formació, per a poder entendre bé les tecnologies a utilitzar, això ha fet que les altres fases comencessin una mica més tard del que es va planificar. També la implementació s'ha allargat per a diverses qüestions, des de buscar solucions a la capa de presentació a tornar a fer un estudi de com pujar un projecte *Scriptcase* al servidor de producció un cop l'hem finalitzat. Per altre banda, en la fase d'especificació s'ha

reduït temps. Tot això fa un retràs de 25 hores en el projecte, la qual cosa no ha suposat un risc. També s'han modificat les iteracions:

Iteració	Funcionalitats
1	Muntar tots els sensors. Programar el microcontrolador configurant-lo per a que es connecti a internet i que reculli les dades dels sensors que hem muntat cada 50 segons per enviar-les.
2	*Backoffice*: iniciar sessió, creació microcontroladors, sensors, mesures, alarmes, accions, veure historial alarmes, validar-les, creació usuaris.
3	Implementació *webservice* per rebre les dades dels sensors. Veure que s'estan enviant correctament Que podem visualitzar les dades des del *backoffice* Que es llencen les alarmes si els valors estan entre els llindars establerts
4	Implementació de les accions varies. Enviar correus. Enviar missatges amb la plataforma sit Mobile. Trucar a telèfons connectant-nos amb el servidor *Asterisk*.
5	Muntar *led* connectat a relé. Modificar microcontrolador per a que activi/desactivi el *led*.
6	Implementar pantalla per veure les dades dels sensors d'una ubicació. Implementar pantalla per veure en un gràfic les dades de les últimes 6 hores d'un sensor determinat. Fer una pantalla similar per rangs de dates de qualsevol sensor.
7	Implementar els triggers en totes les taules que volem tenir-hi un control.

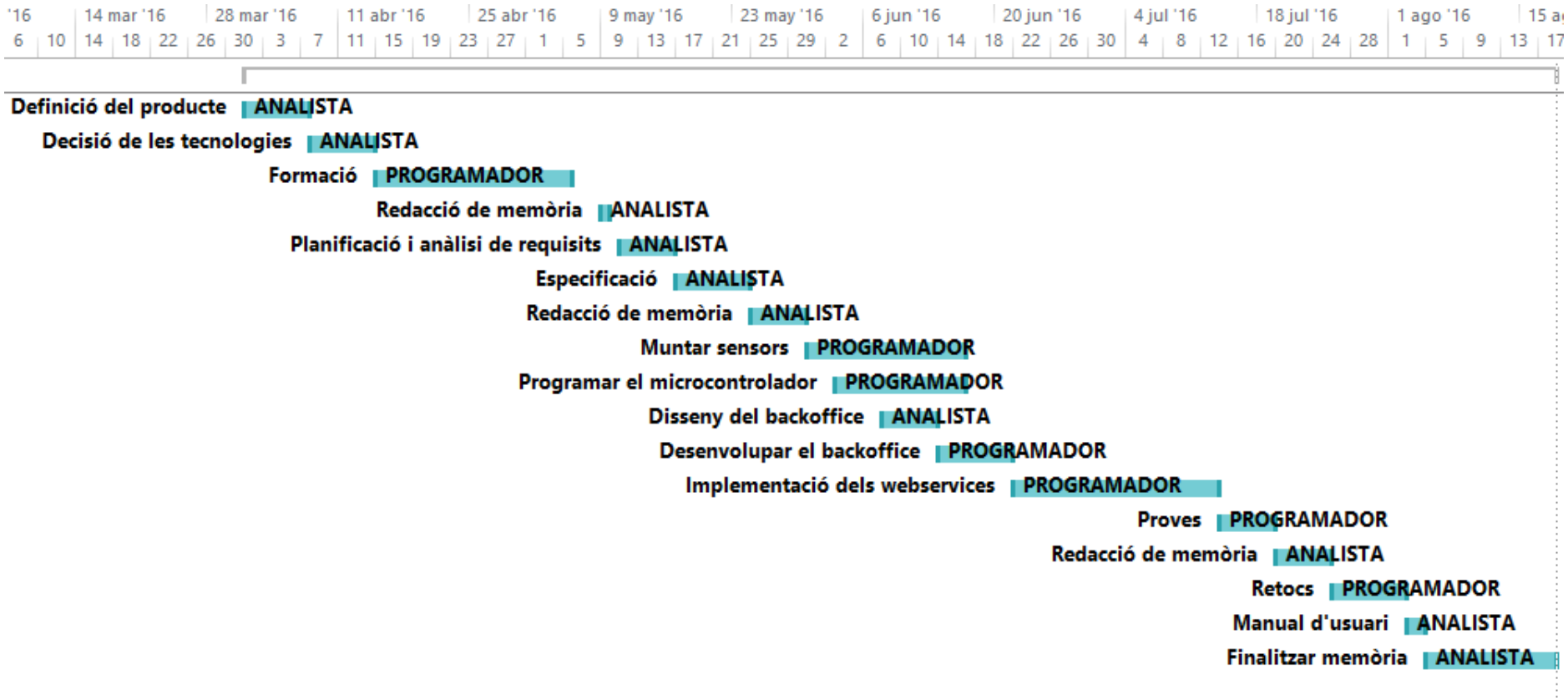
A la primera iteració, a raó de que encara no estaven muntats tots els sensors, es va decidir avançar en temes com la creació de les pantalles (*Backoffice: Scriptcase*) i creació de la base de dades, això fa que la primera iteració acabi ben bé dues setmanes més tard del que es va planificar.

El motiu pel qual s'ha allargat la primera iteració, és a més a més de la part de formació que pertoca als microcontroladors i sensors, que ens ha dut més de l'esperat, que finalment hem

decidit posar també un sensor de portes magnètiques i poder canviar l'estat d'un *led* mitjançant un relé. A més d'això, també hem hagut de substituir el sensor SHT15 pel DHT11, a causa d'una averia.

Per tot això, la segon iteració l'hem hagut de començar en paral·lel amb la primera.

A continuació el Diagrama Gantt del projecte amb la planificació real:



12.1 Anàlisis de costos

12.1.1 Recursos humans

Al ser un projecte final de carrera hi ha hagut una sola persona treballant sota la supervisió del tutor de la universitat. No obstant, s'ofereix una simulació dels costos amb un equip real, que es compondria per un analista i un programador.

Recurs	Salari
Analista	40€/h
Programador	20€/h

D'acord amb l'assignació de recursos humans en cada tasca i la planificació temporal, es pot fer un càlcul del cost de desenvolupar el projecte. El calendari té dies de 8h.

Programador	395hores	7900,00€
Redacció documents	25h	300,00€
Manual usuari	10h	200,00€
Implementació	200h	300,00€
Proves	40h	600,00€
Retocs	30h	400,00€
Formació	90h	1500,00€
Analista	235hores	9400,00€
Especificació	30h	800,00€
Redacció documents	20h	600,00€
Definició del projecte	25h	800,00€
Disseny de l'aplicació	40h	1600,00€
Decisió tecnologia	40h	600,00€
Finalitzar documents	30h	800,00€
Redacció documents	25h	400,00€
Planificació i anàlisis	25h	600,00€

Cost recursos humans = cost programador + cost analista = 7900€ + 9400€ = 17300€

12.1.2 Recursos materials

A més a més, hem de tenir en compte el cost del material utilitzat per muntar els sensors, junt amb el portàtil que hem utilitzat per desenvolupar el projecte.

Recurs	Preu
Arduino	25€/h
Cablejat vari	7€/h
Sensor SHT15	17€
Sensor DHT11	5,5€
Sensor llum	3,5€
Sensor porta magnètica	7€
Relé	14,5€
Protoboard	20€
Led	0,5€
1 Llicència Scriptcase Enterprise Edition	563€
Paquet 500SMS Sit Consulting	36,3€
Portàtil Asus X554L	700€

Cost recursos materials = cost material = 1399,3€ (22)

12.1.3 Cost total

La suma de les hores de feina i els recursos materials: Cost = 17300€ (Cost recursos humans) + 1399,3€ (Cost material)= **18699,3€.**

13 Conclusions i treball futur

Tot i que al resultat final encara li falten algunes millores (esmentades a continuació) aquest projecte ha sigut satisfactori perquè desenvolupar aquesta aplicació m'ha permès comprovar que he assolit les competències professionals relacionades amb aquesta tasca pròpies d'un enginyer superior en informàtica. La conclusió més important es l'assoliment de les tasques principals de l'aplicació.

A més a més, he pogut comprovar que tot i no tenir quasi experiència en l'àmbit electrònic, he sigut capaç de anar muntant poc a poc cada sensor amb el que m'anava trobant, gràcies a la gran documentació que he trobat per internet.

Altres conclusions són:

- El desenvolupament de l'aplicació ha requerit d'un autoaprenentatge de les tecnologies utilitzades, el que li dona un valor afegit al projecte.
- La importància de l'anglès per aprendre ultimes tecnologies, ja que molta de la informació útil trobada era en anglès.
- El posar en comú els coneixements adquirits amb assignatures com ES1, ES2, DABD, GSI... m'han permès planificar, especificar i dissenyar l'aplicació de manera correcte i estructurada. Del projecte en queda la satisfacció de la creació d'un projecte des de zero, de poder experimentar, a petita escala, tots els processos que calen per a assolir uns objectius, des de la idea inicial a un resultat funcional.

14 Bibliografia

1) Extret de www.pigchamp-pro.com/servicios/gestion-de-datos-y-analisis-de-productividad/sensores-farmsense/

2) Extret de http://www.serida.org/proyectodetalle.php?id=352

3) Extret de http://www.agrodigital.com/PlArtStd.asp?CodArt=83750

4) Extret de vikipedia: https://ca.wikipedia.org/wiki/Microcontrolador

5) Extret de : http://www.instructables.com/id/Arduino-Tutorial-0-Introducci%C3%B3n/step3/Qu%C3%A9-partes-componen-el-Arduino/

6) Extret de: http://www.nextiafenix.com/arduino-vs-pic-la-gran-batalla/

7) Extret de: http://www.buenastareas.com/ensayos/Diferencias-Entre-Sensores-Anal%C3%B3gicos-y-Digitales/632269.html

8) Extret de vikipedia: https://ca.wikipedia.org/wiki/Sensor

9) Extret de: http://medirtemperatura.com/sensor-temperatura.php

10) Extret de vikipedia: https://ca.wikipedia.org/wiki/Sensor#Sensors_d.27humitat

11) Extret de vikipedia: https://ca.wikipedia.org/wiki/LDR

12) Extret de : https://www.tecnoseguro.com/faqs/alarma/%C2%BFque-es-un-detector-magnetico-de-apertura.html

13) Extret de vikipedia: https://ca.wikipedia.org/wiki/Placa_de_proves

14) Extret de vikipedia: https://ca.wikipedia.org/wiki/Rel%C3%A9

15) Extret de: http://elpaladintecnologico.blogspot.com.es/2013/03/scriptcase-generador-de-php.html

16) Extret de vikipedia: https://ca.wikipedia.org/wiki/Asterisk

17) Extret de: https://github.com/PHPMailer/PHPMailer

18) Extret de: http://www.rgraph.net/

19) Extret de https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

20) Extret de https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

21) Microcontrolador i sensors:
http://www.ftdichip.com/FTDrivers.htm
https://www.arduino.cc/en/Main/Software
https://www.sparkfun.com/datasheets/Sensors/SHT1x_datasheet.pdf
Sensor temperatura i humitat SHT15

https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity_Sensors/Sensirion_Humidity_Sensors_SHT1x_Datasheet_V5.pdf

Sensor llum LDR:

http://fabricadigital.org/leccion/sensor-de-luz-ldr-y-divisor-de-tension/

Polaritat i connexió d'un Led:

http://www.tuelectronica.es/tutoriales/electronica/como-conectar-un-diodo-led.html

22) Material, extret de diverses fonts:

https://www.amazon.es/Piezas-Fotorresistencia5-Photoresistor-Sensor-Arduino/dp/B00RQX3R6I/ref=sr_1_1?ie=UTF8&qid=1470927781&sr=8-1&keywords=sensor+luz+arduino

https://www.amazon.es/M%C3%B3dulo-Humedad-Temperatura-Digital-Arduino/dp/B01GFXDVV0/ref=sr_1_8?ie=UTF8&qid=1470927812&sr=8-8&keywords=SHT15

http://www.diotronic.com/sensor-digital-humedad-relativ_27393/

http://www.diotronic.com/int-mag-na-nc-max-abert-15mm_2061/

http://www.diotronic.com/modulo-4-reles-12v_27511/

http://www.diotronic.com/placa-protoboard-prof-mediana_967/

http://www.diotronic.com/065228-led-rojo-3mm-intermiten_27209/

https://openwebinars.net/tutorial-arduino-entradas-analogicas-y-digitales/

Cablejat *Arduino*:

http://www.miniinthebox.com/es/cables-macho-a-macho-para-tableros-de-circuitos_p364598.html

https://giltesa.com/2016/03/29/3-curso-iot-con-arduino-y-esp8266-wifi-programacion-arduino-pro-mini

https://www.amazon.es/dp/B00INWWVKY/ref=asc_df_B00INWWVKY35519103/?tag=googshopes-21&creative=24526&creativeASIN=B00INWWVKY&linkCode=df0&hvdev=c&hvnetw=g&hvqmt=

Llicències Scripcase:

https://www.scriptcase.net/es/comprar-scriptcase/?nm_ses=1789f4n5dpidjtuc0ruq74kt61kpd9q6

Paquet 500 missatges amb Sit Consulting:

https://www.sitmobile.com/sadpe-by-sitmobile/

15 Programari

Scriptcase:

http://www.scriptcase.net/download-scriptcase/

http://www.scriptcase.net/docs/es_es/v81/instalacion-scriptcase/instalacion-manual-scriptcase

Asterisk:

http://www.asterisk.org/downloads

Arduino IDE:

https://www.arduino.cc/en/Main/Software

Phpmailer

https://github.com/PHPMailer/PHPMailer

RGraph

http://www.rgraph.net/

Notepad++

https://notepad-plus-plus.org/download/v6.9.2.html

16 Glossari

Arduino: És una placa de circuit imprès simple basada en un microcontrolador de codi obert amb l'objectiu de fer més simple i accessible el disseny de circuits electrònics amb microcontroladors.

Backoffice: És el conjunt d'activitats de suport al negoci. És la part de las empreses que realitzen las tasques destinades a gestionar la pròpia empresa i que no tenen contacte directe amb el client.

Led: És un dispositiu semiconductor que emet llum incoherent d'espectre reduït quan se'n polaritza de forma directa i és travessat per corrent elèctric

Scriptcase: Eina generadora de codi PHP per desenvolupar entorns de *backoffice*, de forma senzilla i ràpida.

Asterisk: És una aplicació de programari lliure (sota llicència GPL) que proporciona funcionalitats d'una central telefònica.

Relé: És un mecanisme que permet modificar l'estat d'un commutador elèctric gràcies a l'electricitat

Protoboard: És una placa d'ús genèric reutilitzable o semi permanent, usada per construir prototips de circuits electrònics amb soldadura o sense. Normalment s'utilitzen per a la realització de proves experimentals.

Curl: És un projecte de software consistent en una biblioteca (*libcurl*) i un intèrpret de comandes (*curl*) orientat a la transferència de fitxers.

Form: Formulari d'introducció de dades, utilitzat en Scriptcase.

Grid: Llistat dels registres d'una taula, utilitzat en Scriptcase.

FTP: És un programari estandarditzat per enviar fitxers entre ordinadors amb qualsevol sistema operatiu.

Webservice: És una col·lecció de protocols i estàndards que serveix per intercanviar dades entre aplicacions.